THE REQUIREMENTS MANAGEMENT PRACTICES: A STUDY AT UUM IT

ENAS FADHIL ABDULAMEER

Supervisor
Dr. Nor Laily Binti Hashim

SCHOOL OF COMPUTING
UUM COLLEGE OF ARTS AND SCIENCES
UNIVERSITI UTARA MALAYSIA
2015

Permission to Use

In presenting this dissertation in partial 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 dissertation 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 dissertation 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 dissertation.

Requests for permission to copy or to make other use of materials in this project dissertation, 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

Abstract

Requirements engineering is a main process in software engineering that focusing on development and managing the user requirements. One of the requirements engineering activities is requirements management. It plays an important role when it comes to the support of product development teams. Despite this, there is a lack of practice in requirements management activity in the software project development. Malaysian software markets are still facing several problems in requirements management practices such as requirements quality, requirements inadequately, and identification of requirements; with limited studies that address it. In this study, UUM IT as computer services provider in a local universities in Malaysia is design as case study, to represent as one organization in Malaysia software markets. This study aims to investigate the current situation for the requirement management in UUM IT, and assess the relationship CMMI level 2 with the requirements management practices in UUM IT. This study adopted mixed method through used questionnaire with the UUM IT team, as well as, interviews with managers of UUM IT for more reliability. The outcome of study showed that the UUM IT are used requirements management activities but there is a need for more attention and improve. Moreover, the study proposes CMMI appraisal method to enhance the performance of software development team.

Keywords: requirements engineering, CMMI level 2, UUM IT

Acknowledgement

In the name of God, the Most Gracious, the Most Merciful.

All praises and thanks to the Almighty, Allah (SWT), who helps me to finish this study, Allah gives me the opportunity, strength and the ability to complete my study for a Master degree after a long time of continuous work. No volume of words is enough to express my gratitude towards my guides, Dr. Nor Laily Hashim without her knowledge and assistance plus her recommendations in this study would not have been successful, She has helped me to explore this topic in an organized manner and provide me with all the ideas on how to work towards a research-oriented venture.

Finally, it would not be possible for me to complete the study and this project without the help of Allah and the support and encouragement from my family and friends. First and foremost, my gratitude goes to my mother to motivate me and for her endless support for me, may Allah bless her. To my husband for supporting me and has a great influence to finish my master study. To my great friends, thanks for standing beside me and giving support in all periods of study.

Thanks for all who have helped or contributed to assist me in completing justified this study.

Table of Contents

List of Tables

List of Figures

List of Appendices

CHAPTER ONE
INTRODUCTION

1.1 Introduction

The software industry is one of the fastest growing industries in the world due to the huge and increasing demand for software applications. The ways of software development can be by standards, needs and company's circumstances. Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (Hoda, Noble, & Marshall, 2012; De Lemos, et al., 2013; Fitzgerald & Stol, 2014; Šmite, Wohlin, Galviņa, & Prikladnicki, 2014). However, the development of the software has become a challenge in order to support the complexity in this domain. Although there are various ways of software development; the weaknesses from the management perspective in software development are always being criticized (Shahid, Ibrahim, & Mahrin, 2011; Osman, 2013).

Requirements Engineering (RE) is a main process in software engineering that is focusing on development and managing the user requirements (Laplante, 2013; Katina, Keating, & Ra'ed, 2014); it is essential during software development in order to ensure the successfulness of software development projects. Theoretically, Requirements Management (RM) is one of the RE activist that focuses on managing requirements over the entire software development (Shahid, Ibrahim, & Mahrin, 2011). According to (Zainol & Mansoor, 2008), there is a lack of practices of RM activist during software project development. With the intention to guarantee the quality of a software product,

RM should be considered during a software development. In addition, the quality of software requirements specification can be compromising and this lead to emphasis on embedding the engineering disciplines into RE processes including the RM practices by employing the best practices, techniques and methodology (Damian et al., 2007; Sommervile, 2010)

In previous years, plenty of RE models, best practices, and appropriate techniques have been established to deliver RE processes. In addition, mach empirical research have been verified that obligating a precise RE process, and embracing appropriate RE techniques together with practices has significant impact on software quality and the overall savings of project cost compared to those projects that did not have well defined process models and did not use suitable techniques as well as practices (Sommervile, 2010; Horkoff & Yu, 2014).

Nowadays, issues in managing requirements become a global issue that indicate the cancellation of software or failure (Ahonen & Savolainen, 2010; Cerpa & Verner, 2009; Dalcher, 2009). In the context of Malaysian software market, this issue remains unknown with very limited studies that address it. In order to reduce this gap, this study is intentionally to investigate the requirements management practices among the software developers at Universiti Utara Malaysia (UUM) IT. Although this study is only focusing on UUM IT, it also contributes to the whole requirements management practices in the Malaysian software industry.

(UUM) is one of the public universities located at the northern part of Malaysia. As an eminent management university, UUM IT is a center that is established with the intention to support and provide services to students, staff and UUM community in completing their daily business activities (UUM, 2015). Presently, UUM has its own systems to handle academic, student affair, bursary and many other related system; which are constructed and maintained by the UUM IT. Thus, the UUM IT needs to deliver high quality software within the budget and on specified time. In order to support UUM IT in meeting the dateline for software delivery, the requirements management practices should be employed during software development. Thus, this study aims to explore the current requirements management practices during software development projects at UUM IT.

1.2 Statement of Problem

Software project requirement engineering is considered as a vital process in software development (Hofmann & Lehner, 2001; Quispe, Marques, Silvestre, Ochoa, & Robbes, 2010; Harman, McMinn, Souza, & Yoo, 2012; Kassab, Neill, & Laplante, 2014). In most cases, RE practices which involved requirements development and requirements management are greatly emphasized whenever an organisation adopts quality best practices in their processes. However, there are many models that can be used to improve the practices and process, such as Capability Maturity Model Integration (CMMI) (Kim & Grant, 2010; Samalikova, Kusters, Trienekens, & Weijters, 2014; Mahmood, Dhakal, Wiewiora, Keast, & Brown, 2015). These models have embedded the use of requirements development, and RM process areas. Ahonen and Savolainen

(2010) and KarimJallow, Demian, Baldwin and Anumba (2014) stressed that the success or failure of the software product depends largely on the RM process.

In project development, RM faces many challenges. Firesmith, (2007) and Basili et al. (2013), refer that the requirements quality is one of the main challenges faces in RM. In quality challenge, there are many issues that affect on requirements, such as incompleteness of a requirements specification, vague requirements statements, and redundant requirements statements (Mu, Hong, Jin, & Liu, 2013). Therefore, verification of quality requirements depends on good management.

Pohl, (2013) confirmed that most of software projects manage their requirements inadequately by storing their requirements in documents or spreadsheets. There are many type of requirements and the requirements are kept in different media separately. This lead to difficulty in accessing these requirements and most of the requirements are missing important metadata, for example status, priority, type and many others (Firesmith, 2007; Li, Brown, Hayes & Truszczynski, 2014). Additionally issue regarding requirements is to detect errors in requirements in the early phase of software development project in order to reduce the cost of fixing the error at the later phase. Fixing requirements problems may require rework of the system design, implementation and testing (Charette, 2005; Nasir, & Sahibuddin, 2011). Moreover, there is an issue with untrained and unskilled requirements engineers that lead to insufficient access to stakeholders and supplementary sources of requirements (Firesmith, 2007; Crespo & Ruiz, 2012).

Most the projects failed because of poor identification of requirements (El Emam & Birk, 2000). Kumar and Kumar (2011) stated that poor requirements leads to increase of overall cost, decrease quality of the system or system frailer. In Malaysia, the development of software industry has attracted researchers with particular interest in issues related to requirements management of the software projects (Abdul Manaf 2006; Othman, Zain & Hamdan, 2010; Rahman et al., 2014). Where these studies have focused on investigate RE issues facing the stakeholder perspectives, as well as, to understand the impact that is brought by the dynamic environment of projects developed; despite the importance of the RM, but it was part of the study. However, these studies are focused on investigating the current situations of the RM practices in general in the Malaysian companies. However, in this study the researcher focuses on the actual implementation of RM practices in an organization.

1.3 Research Questions

The research questions are:

1. What are the current RM practices in UUM IT?
2. What are the issues that affect the adoption of the appropriate RM practices in UUM IT?
3. What is the relationship of RM practices with CMMI level 2 at UUM IT?

1.4 Research Objectives

The research objectives are:

1. To identify the current RM practices in UUM IT.
2. To identify issues that affect the adoption of the appropriate RM practices in UUM IT.
3. To assess relationship of RM practices with CMMI level 2 at UUM IT.

1.5 Scope of the Study

RM is managing the different levels, types and attributes of requirements throughout the software life cycle. This study aims to investigate the requirements management practices issues and assess against the CMMI level 2. The scope of this study is focusing on UUM IT. UUM IT is one of oldest IT center for public universities in Malaysia, where it was established in 1989. This centers is supervising on a big campus to provide a computing and network services at UUM. Thus, the developments of software are update. On other hand, cost and time play an important reason to select UUM IT as a case study for this empirical study.

1.6 Significance of Research

The basic idea of RM of the software projects is to ensure success of a project development. The current study is attempted to investigate the current situation of the RM practices in the UUM IT. The significance of this study is to assist management of UUM IT for designing and implementing relevant measures to improve software

projects delivery in UUM IT. Moreover, to provide the accurate information about the current situation by benchmarking the RM practices with CMMI level 2.

1.7 Organization of the Project

First chapter captures the practical gap; through presented the background of the study whereby the problem of the research is put into light; the objectives and research questions are set. Moreover, this chapter describes the research significance and scope.

Chapter Two provides a review of literature related to the RE and RM, as well as, explanation the importance of CMMI in detail.

Chapter Three emphasizes on the research methodology which is developed in three stages, as well as the design of the questionnaire were clarified. Chapter Four presents the analysis of the outcome for the questionnaire. Lastly, chapter five provides the concluding remarks on the study, as well as suggestions and recommendations for future research.

1.8 Summary

In this chapter, brief introduction has been introduced about the RE for development software projects, focusing on the RM practices to improve the software project. The problem statement identified the issues that effect on RM, as well as, the current situation of RM for the software project in Malaysian companies needs to investigate. At the end of chapter are presented a clear view of the scope and the significance of the study.

CHAPTER TWO
LITERATURE REVIEW

Software applications are getting important these days as software are being used to support our daily life as in washing machine, components of a car and almost everywhere in our life. Software becomes important in financial, health care, education and many other domains. Thus, developing software become complex and difficult activity as it need to satisfy all the clients need in every domain. A software application that meets the expectations of the clients, which could be delivered on time, on budget should be developed on the basis of correct and appropriate requirements.

A requirement is a capability that a software system must provide, or an attribute that this system must have in order to resolve issue or accomplish certain objective within a specific domain (McLeod & McDonell, 2011). According to this definition, it is clear that a software system is embedded in an application domain, so its usefulness depends on the problems that it can solve and on the objectives that it can achieve in such domain. However, to get the correct and appropriate requirements is a difficult task. This is also been highlighted by (Hansen, Berente, & Lyytinen, 2009) that requirements are ultimately the most challengers in designing the software intensive system. Therefore, this chapter provides a review of the relevant literature upon which this study. In the first section, an overview of requirements engineering is addressed. This is followed by overview on requirements management, which is reviewed to understand

the activities for requirements management. In the third section, CMMI is discussed to shed light into the level 2. Finally, a summary of this chapter was displayed in section 2.4.

2.1 Requirements Engineering

The most data-intensive software development activities lie in requirements engineering (RE) where stakeholders are determined, problems are explored, and goals are defined (Gotel, Marchese& Morris, 2008). Not only does RE involve the identification of the diverse stakeholder concerns and the complex environmental constraints, critical decisions are also made in RE when business objectives are transformed into technical specifications, when conflicting viewpoints are negotiated into an agreed upon action plan, and when reasoning are performed to enable revisions without incurring the serious time or budget overturns. Keeping requirements on track thus requires a disciplined approach to handling the massive, heterogeneous, and dynamic volumes of information (Reddivari, 2013).

RE is the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction. It is, widely recognized as the first phase of software engineering process, is considered the key task of software development (Asghar & Umar, 2010; Wahono, 2003). RE is an organized approach which the software engineer gather, analyze, evaluating and documenting the requirements in order to implement them during software development. RE activities cover the entire

system and software development life cycle. The RE process is an iterative process which also indicates that the RM is understood as an aspect of RE process (Pohl, 2010; Robertson & James, 2012; Sommervile, 2010). Traditionally, RE is performed at the beginning of the system development life cycle (Thayer & Dorfman, 1990). Nevertheless, in sophisticated system development, it might takes months and years in developing a stable requirements and this is impossible to be implemented in practice (Robertson & James, 2012). Consequently, RE is an incremental and iterative process, performed in parallel with other system development activities such as design, coding etc.

Many recent studies recommend better requirements engineering activities to address problems in system development programs (Blanchard & Fabrycky, 2010; Grenn, 2013), and the companies have implemented major changes in its systems acquisition practices to place greater emphasis on early SE to reduce the negative effects of requirements volatility later in the program. However, the field of RM has sub-domains. According to Wiegers (2003) splits the software RE into two domains which are requirements development and requirements management, each of domain was divided in four parts, as shown in Figure 2.1.

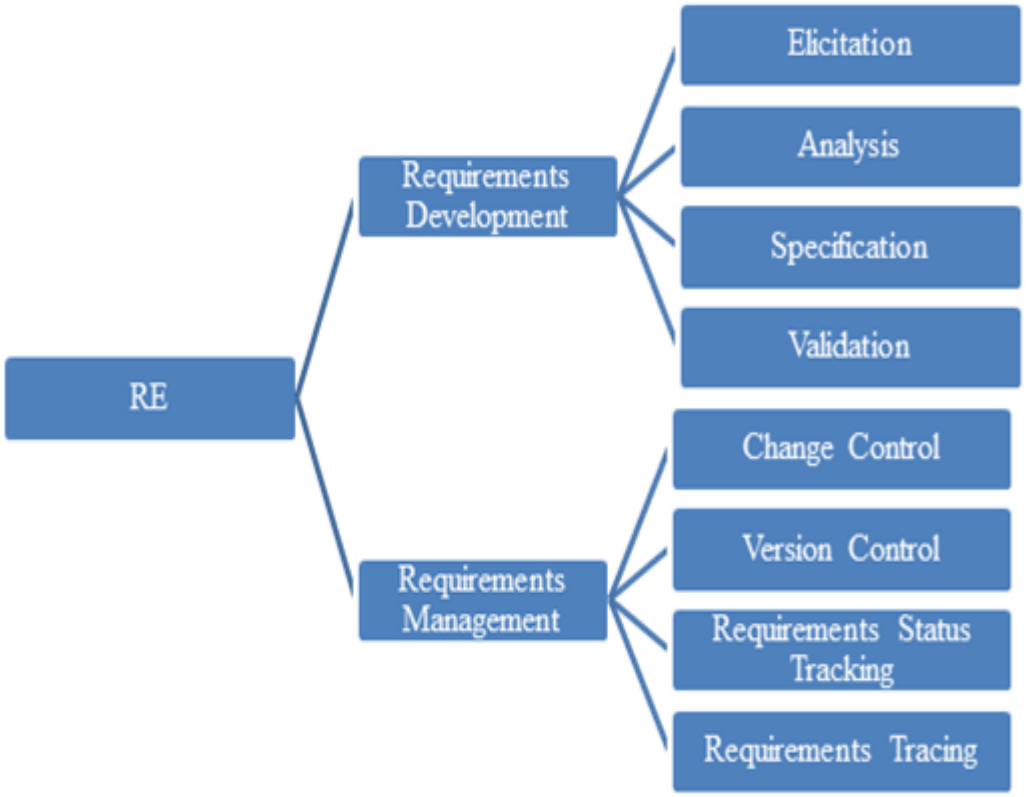

Figure 2.1: Requirements engineering sub-domains

Source: (Wiegers, 2003)

Requirements development is one of the main domain or RE. According to Abran and Moore, (2001) referred that the requirements development was consists of four main activity which are requirements elicitation refers to all of the activities involved with discovering requirements, such as interviews, workshops, document analysis, prototyping, and others. Requirements analysis, which is which is involves reaching a richer and more precise understanding of each requirement and representing sets of requirements in multiple ways. Requirements specification involves representing and storing the collected requirements knowledge in a persistent and well-organized fashion.

Lastly, requirements validation refers to set of requirements information that will enable developers to build a solution that satisfies the business objectives. However, the current study focuses on the requirements management, which is addressed in detail next section.

2.2 Requirements Development

The process of requirements development is often implemented with the process of requirements management. In fact, the requirements development is concerning with the level 3 process, as well as, it is a best practice for defining customer and product requirements. According to O'Regan (2010); Rosenkranz, (2012) the requirements development is concerned with best practice for defining and documenting the customer and product requirements. It is also concerned with analyzing and validating the requirements. It includes practices for eliciting customer and product requirements as well as analyzing and validating the requirements, these activities will be discussed successively.

2.2.1 Requirements Elicitation

The requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders (Wiegers & Beatty, 2013). Elicitation techniques include both facilitated activities, in which you interact with stakeholders to elicit requirements, and independent activities, in which you work on your own to discover information. Facilitated activities primarily focus on discovering business and user requirements. Working directly with users is necessary because user requirements

encompass the tasks that users need to accomplish with the system. To elicit business requirements, you will need to work with people such as the project sponsor. The independent elicitation techniques supplement requirements that users present and reveal needed functionality that end users might not be aware of. Most projects will use a combination of both facilitated and independent elicitation activities.

2.2.2 Requirements Analysis

Requirements analysis involves refining the requirements to ensure that all stakeholders understand them and scrutinizing them for errors, omissions, and other deficiencies. Analysis includes decomposing high-level requirements into appropriate levels of detail, building prototypes, evaluating feasibility, and negotiating priorities (Pohl, 2013). The goal is to develop requirements of sufficient quality and precision that managers can construct realistic project estimates and technical staff can proceed with design, construction, and testing.

2.2.3 Requirements Specification

The requirements specification refers to the document requirements of different types in a consistent, accessible, and reviewable way that is readily understandable by the intended audiences. The developer can record the business requirements in a vision and scope document (Wieringa, 2014). User requirements typically are represented in the form of use cases or user stories. Detailed software functional and nonfunctional requirements are recorded in a software requirements specification or an alternative repository, such as a requirements management tool.

2.2.4 Requirements Validation

Validation ensures that the requirements are correct, demonstrate the desired quality characteristics, and will satisfy customer needs. Requirements that seem fine when you read them might turn out to have ambiguities and gaps when developers try to work with them (Lee, Min, Cho & Lim, 2012). The developer must fixit these errors when the requirements are for final system testing or to serve as a reliable foundation for design and for user acceptance testing.

2.3 Requirements Management

Requirement management as a process area refers to the process of defining and documenting the project and product features and functions needed to fulfill stakeholders' needs and expectations. The project's success is directly influenced by the care taken in capturing and managing these requirements. The purpose of requirements management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.

The specific goal in the assessment determines whether requirements are managed and whether there are inconsistencies with the project plans and work products. The specific goal also determines whether these inconsistencies are identified. The specific practices included: Developing understanding with the requirements providers on the meaning of the requirements; obtaining commitment to the requirements from the project participants; managing changes to the requirements as they evolve during the project;

and identifying inconsistencies between the project plans and work products and the requirements.

Requirements management plays an important role in the development of software systems. Being the first step in the process of software engineering, the effort has potential to shape the direction for all subsequent project activity. Taken together, requirements by their nature provide us a model of the system to be built. A project that meets its requirements is by definition a success. RM, then, starts with the definition of requirements and continues through the project, culminating in the acceptance o f the product against the requirements.

In the previous work, RM is broadly defined in many ways (Leffingwell&Widrig, 2003; Sommervile, 2010; KarimJallow, Demian, Baldwin &Anumba, 2014). Dutoit and Paech, (2000) defined RM as "The systematic process of organizing and storing relevant information about requirements, while ensuring requirements traceability, and managing changes to these requirements during the whole lifecycle of the information system". As in many other management disciplines, the essence of requirements management is simple: establish expectations, meet expectations. But the process becomes more complex in real-world application. A list o f tasks associated with effective requirements management as illustrated in Table 2.1 (Persse, 2001).

Table 2-1: List of tasks associated with requirements management

Number	Tasks
1	Define and communicate what is wanted.
2	Provide traceability to outside documents.
3	Apply requirements to the solution.
4	Optimize the product before commitment.
5	Drive the design and the implementation.
6	Manage change, problem reports, and suggestions.
7	Manage the partitioning of work to specialists.
8	Test and validate the finished product.
9	Control iterative developments.
10	Manage project milestones.
11	Manage interfaces with external systems.

This table showed that the requirements management, then, is broad-based. It affects every aspect of the project development lifecycle. And it involves, to a large or small extent, every member of the development effort, from the user to the testers. For this reason, its impact on project success is also broad. When managed well, requirements efforts can greatly aid in the development process; when managed poorly, deep and significant problems may arise.As a result, the object of requirements management is not to stifle change or to make it difficult. It is to anticipate and accommodate the very real changes that you can always expect so as to minimize their disruptive impact on the project (Wiegers & Beatty, 2013). The RM have four main activates that described in details next section.

2.3.1 Requirements Management Activities

Requirements management includes all activities that maintain the integrity, accuracy, and currency of requirements agreements throughout the project. According to Wiegers and Beatty (2013) there are four main activates clarify the role of RM in software development; which are version control, change control, requirements status tracking, and requirements tracing; each activity was determined through some of tasks as described in Figure 2.2.

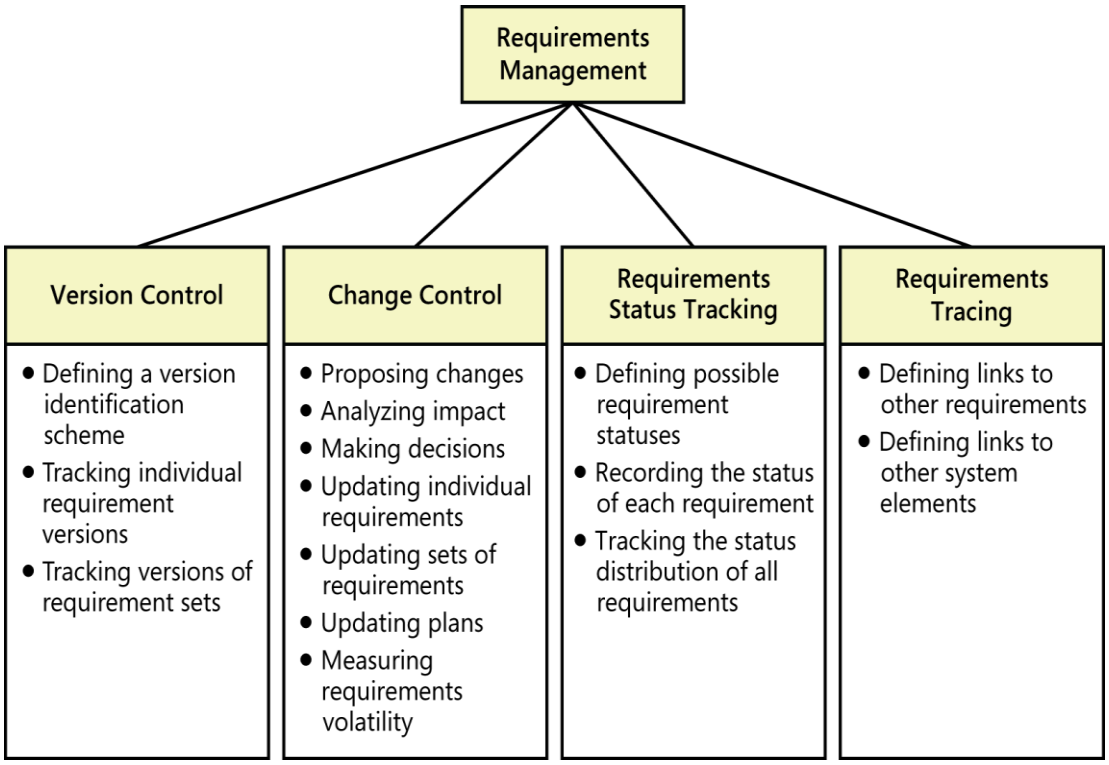

Figure 2.2: The Main Activities of Requirements Management

Source: (Wiegers & Beatty, 2013)

2.3.1.1 Version control

Version control refers to the unique identifying different versions of an item, applies at the level of both individual requirements and requirements sets, most commonly represented in the form of documents. Begin version control as soon as you draft a requirement or a document so you can retain a history of changes made. Based on Wiegers and Beatty, (2013) that the Version control have several tasks most be considered when implementing the project. Table 2.2 illustrated the tasks of version control.

Table 2-2: Version Control Tasks

Number	Tasks
1	Every version of the requirements must be uniquely identified.
2	Every team member must be able to access the current version of the requirements.
3	Changes must be clearly documented and communicated to everyone affected.
4	Permit only designated individuals to update the requirements, to minimize confusion and miscommunication.
5	Each circulated version of a requirements document or each requirement in a tool should include a revision history that identifies the changes made, ➢ the date of each change, ➢ the individual who made the change, ➢ the reason for each change.

These RM, tools "version control activity" is able to identify, track and manage the entire history of requirements changes for every defined requirement. The current study

is attempted to identify the extent of use this activity to improve the requirements management practices for software within the environment of UUM.

2.3.1.2 Change control

Managing requirements changes is similar to the process for collecting and making decisions about defect reports. The same tools can support both activities. Remember, though: a tool is not a substitute for a documented process, and neither one is a substitute for appropriate discussions between stakeholders. Regard both a tool and a written process as ways to support these critical conversations. However, software change is not a bad thing; in fact, it's necessary. It's virtually impossible to define all of a product's requirements up front. The world changes as development progresses: new market opportunities arise, regulations and policies change, and business needs evolve. An effective software team can nimbly respond to necessary changes so that the product they build provides timely customer value. An organization that's serious about managing its software projects must ensure that:

- Proposed requirements changes are thoughtfully evaluated before being committed to.
- Appropriate individuals make informed business decisions about requested changes.
- Change activity is made visible to affected stakeholders.
- Approved changes are communicated to all affected participants.
- The project incorporates requirements changes in a consistent and effective fashion.

But change always has a price. Revising a simple webpage might be quick and easy; making a change in an integrated circuit design can cost tens of thousands of dollars. Problems can also arise if a developer implements a requirement change directly in the code without communicating with other team members. The documented requirements then become an inaccurate representation of what the product does. Even a rejected change request consumes the time needed to submit, evaluate, and decide to reject it. Unless project stakeholders manage changes during development, they won't really know what will be delivered, which ultimately leads to an expectation gap and needs to control the change.

Change control is a process that helps a project manager to control business decision that will benefit the client as well as business value in order to monitor the product cost. Requirements development involves activities to elicit, analyze, specify, and validate a software project's requirements, which is depending on several items constitutes a requirements baseline. Requirements baseline is a set of requirements that stakeholders have agreed to, often defining the contents of a specific planned release or development iteration (Mizouni & Lazarova-Molnar, 2012).At the time a set of requirements is baselined; the requirements are placed under configuration (or change) management, which typically following review and approval. However, subsequent changes can be made only through the project's defined change control procedure (Wiegers& Beatty, 2013).

Prior to base lining, the requirements are still evolving, so there's no point in imposing unnecessary process overhead on those modifications. A baseline could consist of some or all the requirements in a particular Software Requirements Specification (SRS), or a designated set of requirements stored in an RM tool, or an agreed-on set of user stories for a single iteration on an agile project. Therefore, storing requirements in an RM tool facilitates the identification of those that belong to a specific baseline and the management of changes to that baseline. This change can be achieved through several approaches as shows in Table 2.3.

Table 2-3: Change control approach

Number	approach
1	By deferring lower-priority requirements to later iterations or cutting them completely.
2	By obtaining additional staff or outsourcing some of the work
3	By extending the delivery schedule or adding iterations to a project
4	By sacrificing quality to ship by the original date

Table 2.3 illustrated four approaches which helps the team or manager to make a change in the requirements baseline. Despite that, there is no single approach is universally correct, because projects differ in their flexibility of features, staff, budget, schedule, and quality (Wiegers, 1996). Thus, Wiegers and Beatty (2013) confirmed that the change should be based on the project's business objectives and the priorities the key stakeholders established during project initiation. No matter how you respond to changing requirements, accept the reality of adjusting expectations and commitments

when necessary. Thus, the present study attempts to investigate the use of approaches of change in the projects by IT professional in the UUM IT, and their interaction with.

2.3.1.3 Requirements Status Tracking

In the software development, it is important to recognize the status of every requirement because it can show up the latest project's evolvement. One of the requirement attributes to manage the project is a status. Tracking status means comparing where you really are at a particular time against the expectation of what "complete" means for this development cycle (Carstens, Richardson, & Smith, 2013). Thus, tracking the status of each functional requirement throughout development provides a more precise gauge of project progress; therefore, Table 2.4 shows several possible requirement statuses to track the project (Wiegers & Beatty, 2013).

Table 2-4: Requirement statuses

Status	Definition
Proposed	The requirement has been requested by an authorized source.
In Progress	A business analyst is actively working on crafting the requirement.
Drafted	The initial version of the requirement has been written.
Approved	The requirement has been analyzed, its impact on the project has been estimated, and it has been allocated to the baseline for a specific release. The key stakeholders have agreed to incorporate requirement, and the software development group has committed to implement it.
Implemented	The code that implements the requirement has been designed, written, and unit tested. The requirement has been traced to the pertinent design and code elements. The software that implemented the requirement is now ready for testing, review, or other verification.
Verified	The requirement has satisfied its acceptance criteria, meaning that the correct functioning of the implemented requirement has been confirmed. The requirement has been traced to pertinent tests. It is now considered complete.

Deferred	An approved requirement is now planned for implementation in a later release.
Deleted	An approved requirement has been removed from the baseline. Include an explanation of why and by whom the decision was made to delete it.
Rejected	The requirement was proposed but was never approved and is not planned for implementation in any upcoming release. Include an explanation of why and by whom the decision was made to reject it.

Classifying requirements into several status categories is more meaningful than trying to monitor the percent completion of each requirement or of the complete release baseline. Update a requirement's status only when specified transition conditions are satisfied. This requirements classifying are helpful for software practitioner to detect the progress of every requirements. Therefore, to identify the mechanism of interaction with the status tracking for the IT professional in UUM IT, section C for the questionnaire can achieve this aim.

2.3.1.4 Requirements Tracing

Requirements tracing or traceability is one the elements of successful requirements specifications because traceability has a link that acts as a path for requirements to move forward or backward, from the requirements source to implementation (Kulak & Guiney, 2012; Pohl, 2010). There are four types of requirements trace links as shows in Figure 2.2 (Jarke 1998; Kragelund, 2012).

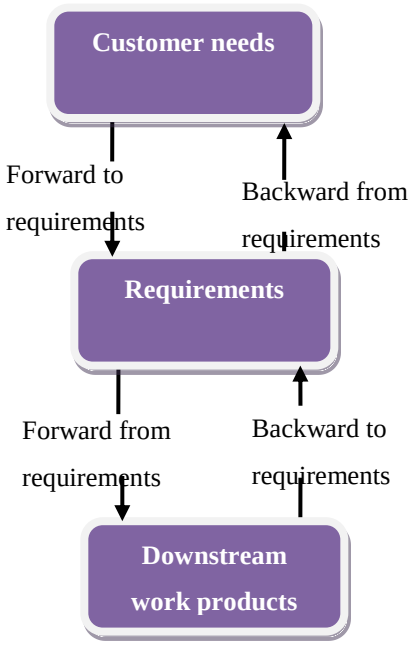

Figure 2.3: Types of requirements tracing
Source: (Wiegers & Beatty, 2013)

Figure 2.3 illustrated that the customer needs are traced forward to requirements, to explain which requirements will be affected if those needs change during or after development. Customer needs could be articulated in the form of business objectives, market demands, and/or user requirements; which gives a confidence that the requirements set has addressed all stated customer needs. Conversely, the programmer can trace backward from requirements to customer needs to identify the origin of each software requirement. If you choose to represent customer needs in the form of use cases.

Moreover, the requirements flow into downstream deliverables during development, the programmer can trace forward from requirements by defining links between individual functional and nonfunctional requirements and specific system elements; which allows determining the satisfied with every requirement by knowing which design components

and code elements address each one. The fourth type of link traces specific product elements backward to requirements to know why each element was created.

Based on above, each requirement is individually identified and personally labeled for easy recognition during entire software development through traceability. However, keeping the link information current as the system undergoes development and maintenance takes discipline and time. If the trace information becomes obsolete, you will probably never reconstruct it. Obsolete or inaccurate trace data wastes time by sending developers and maintainers down the wrong path, destroying any trust the developers might have had in the information. Because of these realities, you should adopt requirements tracing for the right reasons (Saleem, Khan, &Afzal, 2012). Therefore, understand the requirement tracing in the current study is important, the items related with RM activities for section C in questionnaire are illustrated in Table 2.5.

Table 2-5: RM activities for section C in the questionnaire

Item	RM activities	References
Do you carry out a feasibility study before starting a new project?	Requirements Elicitation	(Weber, Curtis & Chrissis, 1995; Anderson & Felici, 2001; Zainol &
Do you reuse requirements from other systems which have been developed in the same application area?	Requirements Elicitation	
Do you define system boundaries?	Requirements Analysis and Negotiation	
Do you use checklists for requirements analysis?	Requirements Analysis and Negotiation	
Do you prioritise requirements?	Requirements Analysis and Negotiation	
Do you perform any risk analysis on requirements?	Requirements Analysis and	

	Negotiation	Mansoor, 2008). Khankaew & Riddle, 2014)
Do you use prototyping to animate / demonstrate requirements for validation?	Requirements Validation	
Do you uniquely identify each requirement?	Requirements Management	
Do you have defined policies for requirements management?	Requirements Management	
Do you use a database to manage requirements?	Requirements Management	
Do you identify global system requirements?	Requirements Management	
Do you identify volatile requirements?	Requirements Management	
Do you record rejected requirements?	Requirements Management	
Do you reuse requirements over different projects?	Requirements Management	
Do you have standards templates / documents for describing requirements?	Requirements Description	

2.4 Capability Maturity Model Integration (CMMI)

Currently, there are several maturity models, standards, methodologies, and guidelines that can help an organization improve the way it does business. However, most available improvement approaches focus on a specific part of the business and do not take a systemic approach to the problems that most organizations are facing. By focusing on improving only one area of a business, these models (CMMI) provides an opportunity to avoid or eliminate these barriers through integrated models that transcend disciplines.

CMMI consists of best practices in software companies. It addresses practices that cover the product's life cycle from conception through delivery and maintenance. There is an emphasis on both systems engineering and software Engineering and the integration necessary to build and maintain the total product (Casallas & Arboleda, 2011). It is an integrated model of many CMMs intended to achieve process improvement. CMMI has two representations Staged and, Continuous (Yadav & Kumar, 2014).

In CMMI, requirements management is intentionally to manage all requirements of products and products components and identify inconsistencies between those requirements and the projects plans and work products (Team, 2006). CMMI has five levels, but with more additions to the processareas. Figure 2.3 shows the levels with their respective process areas. Regnell, Svensson and Wnuk, (2008) refers that the large size of software system ledto the increasing importance of RM to organize the work of developers team, therefore, this study focuses on the level 2 requirements management, which is describing in details next section.

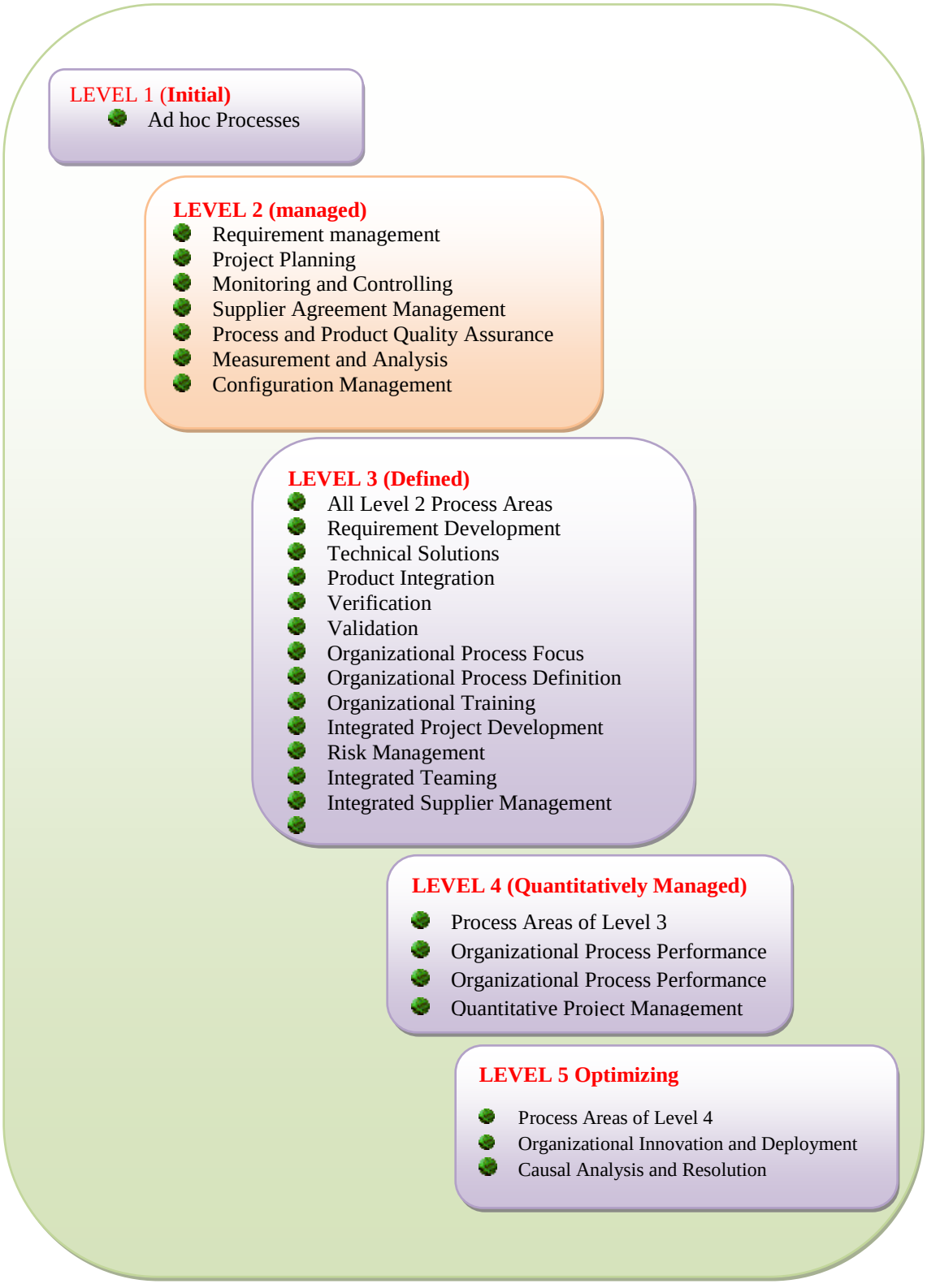

Figure 2.4: CMMI five levels with respective process areas

Source: (O'Regan, 2010).

2.4.1 CMMI level 2

In CMMI models with a staged representation, there are five maturity levels, each a layer in the foundation for ongoing process improvement, designated by the numbers 1 through 5 (O'Regan, 2010). The current study focuses on the second level (Level 2 managed). At maturity level 2, the projects have ensured that requirements are managed and that processes are planned, performed, measured, and controlled. The process discipline reflected by maturity level 2 helps to ensure that existing practices are retained during times of stress.

When these practices are in place, projects are performed and managed according to their documented plans. Furthermore, requirements, processes, work products, and services are managed. The status of the work products and the delivery of services are visible to management at defined points (for example, at major milestones and at the completion of major tasks). Commitments are established among relevant stakeholders and are revised as needed. Work products are reviewed with stakeholders and are controlled (SEI, 2000).Implementing CMMI level 2 aims to institutionalize effective management practices for projects and repeated the practices that were successful in earlier projects(Team,2006).

2.4.1.1 Requirement Management

Requirements management is concerned with ensuring that the project maintains an up to date approved set of requirements throughout the project and ensuring that the project deliverables are kept consistent with the requirements (Casallas & Arboleda, 2011). It is an important area to get right as all project activities are planned from the approved

requirements. The requirements management process area is concerned with best practice for managing the requirements of the project and in identifying inconsistencies between the requirements and the project plans and work products. Its focus is on the activities for managing the requirements as distinct from the activities in gathering the requirements. The requirements may in many cases be incomplete, inadequately documented, or untestable. Changes to the requirements may lead to a high level of re-work or cause major delays to the schedule or major increases in project cost. According to O'Regan, (2010) the specific goals and practices for the requirements management process area are listed in Table 2.6.

Table 2-6: CMMI requirements for requirements management

Specific practice	Description of specific practice/goal
SP 1	Obtain an understanding of the requirements
SP 2	Obtain commitment to requirements
SP 3	Manage requirements changes
SP 4	Maintain bi-directional traceability of requirements
SP 5	Identify inconsistencies between work products and requirements

2.4.1.2 Software Project Planning

The project plan is a stage to describe the approach to the implementation of CMMI level 2. In this stage the timelines and resources required will be scheduling for setting up and training the various improvement teams (O'Regan, 2010). These will include the

definition of processes and procedures, checklists, training material, tools, pilots, and rollout of the new processes and standards (Chrissis, Konrad & Shrum, 2011).

2.4.1.3 Monitoring and Controlling

The project monitoring and control process is concerned with monitoring project execution and taking corrective action when project performance deviates from expectations. They state that the progress of the project should be monitored against the plan and that corrective actions should be taken when progress deviates from expectations (O'Regan, 2010; Chrissis, Konrad & Shrum, 2011). This involves monitoring key project parameters such as budget, effort, and schedule as

2.4.1.4 Supplier Agreement Management

The purpose of this process area is to manage the acquisition of product from suppliers where there exists a formal agreement. It is concerned with best practice for establishing and satisfying supplier agreements and includes practices to select suppliers, defining an agreement with the supplier, executing the agreement, and accepting the supplier product. The supplier management improvement team is responsible for implementing the specific and generic goals and practices for this process stage (O'Regan, 2010). The activities associated with supplier selection typically include:

1. Identify candidate suppliers, and determine evaluation team
2. Define evaluation criteria, and issue request for proposal
3. Receive proposals, and shortlist suppliers for presentation
4. Rate suppliers against evaluation criteria

5. Make decision, negotiate agreement and statement of work, as well as, manage the supplier

2.4.1.5 Software Configuration Management

The configuration management process area is concerned with the implementation of best practice for establishing a configuration management system; identifying work products that need to be subject to change control; controlling changes to these work products over time; controlling releases of work products; creating baselines; maintaining the integrity of baselines; providing accurate configuration data to stakeholders; recording and reporting the status of configuration items and change requests; and verifying the correctness and completeness of configuration items with configuration audits (O'Regan, 2010; Gupta & Rao, 2011). The configuration management improvement team is responsible for implementing the CMMI requirements for this process area.

Configuration management allows the orderly development of software, and it ensures that the impacts of proposed changes are considered prior to authorization. It ensures that releases are planned and that only authorized changes to the software are made (Gupta & Rao, 2011). The integrity of the system is maintained and the constituents of the software system and their version numbers are known at all times.

2.4.1.6 Process and Product Quality Assurance

This stage concerned with the work products being produced by the projects, as well as, providing visibility to management on the processes is followed. O'Regan (2010) refers that this process area is "concerned with the implementation of best practice to plan and conduct audits. It includes planning and conducting the audits; documenting and reporting the results to managers and affected individuals; assigning audit actions to individuals or groups to address identified non-compliances; and tracking the audit actions to completion". The implementation of the process and product quality assurance process may be done by the first teams to be set up in the initiative, which the CMMI project manager have a clear view on how this independent function should be implemented.

2.4.1.7 Measurement and analysis

Management information needs have supported through the implementation of best practice for measurement. This measurement involves the needs of management information, Identify the objectives of measurement, specification of measures and implementation, and the results and analyzing. On other word, once the measurement objectives, measures, data collection, storage, and analysis procedures have been specified, the projects start collecting and analyzing measurement data, storing, and communicating the results. The typical deliverables for this process area include a process map, guidelines, templates, and a checklist. The process map provides an abstract summary of the activities involved; the procedure and guidelines provide the details behind the process map (Kähkönen & Abrahamsson, 2004; O'Regan, 2010).

Based on above, this study attempts to investigate the familiarity develop teams of UUM IT with the requirement management CMMI activities, therefore, section D for the questionnaire can achieve this aim.

2.5 Related Work

Anderson and Felici (2001) stated that, Requirements Evolution represents one of the major problems in developing computer-based systems. Anderson and Felici (2001) aimed to investigations of live industrial contexts to devise product oriented approaches supporting Requirements Evolution. This study used mixed methods as the approach for collecting data (qualitative and quantitative). In sum up they identified an empirical framework for the analysis of Requirements Evolution. The framework represents a valuable tool to implement a feedback into software process and product.

The compliance with requirements management determines the success or the failure of a project. Prikladnicki, Audy and Evaristo (2003) therefore stated that, requirements management becomes critical due to the characteristics of the distributed development. Indeed, Prikladnicki *et al.* (2003) carried out study on the geographically distributed environments (Global Software Development) where the main objective for their study is to analyze the RM in geographically distributed environments and to identify the main challenges. The results are based on a case study conducted on at Dell Computers, a multinational organization that has offshore software development centers in Brazil, India and Russia. This study enables a better understanding of the GSD area and the

relationship between the project team, customers and users related to the requirements management phase.

The outcome from this study suggested the necessity to adopt the RM to distributed software development environment. More specifically, this study analyzed two case studies (projects), after the analysis of these two projects, the researchers conclude that to manage requirements in a global software development context can become an arduous task if the process will not be well defined and if the teams will not be previously prepared to work in this scenario. As well as, this study placed a many of lessons that related to requirements management phase, such as:

1. Training the team in soft skills (trust, cultural differences, communication, collaboration, context sharing, knowledge management, etc.) is essential.
2. Work standardization is mandatory.
3. Frequent meetings with people geographically distant are very important to track the project.
4. A well-defined process is a key to success.
5. The use of tools like email, conference calls and video conferences are very important.
6. It's very important to know about the people that you are working, considering the way to communicate, cultural differences, etc.

RM as a systematic approach to identify, document, organize, and track all system's requirements. Cuevas, Serrano and Serrano (2004) conducted a study to provide an accurate picture of the organization's RM process by the use of an assessment methodology based on a two-stage questionnaire. These questionnaires were based on the two practices of the RM process area of the Capability Maturity Model Integration (CMMI). This study found that the alternative assessment methodology based on a two-stage questionnaire that proposed in this article, provide valuable information related to those areas that require prioritize.

In this study two specific practices and three generic practices showed some major problems. These suggest that they need to be a priority for the action plan. Therefore this study is aware that, the identification of the practices that need to be implemented is only the first step of a continuous process and that in order to aim for a successful Software Process Improvement program there is the need to describe how to implement the identified practices.

Evidences suggest that some of the most common and serious problems associated with developing software are related to requirement management. Therefore, Verner, Bleistein, Cerpa and Cox (2006) carried out a study to recognize what RE practices that are actually used and which of these practices lead to development of good requirements. The main method harness for the collected data was quantitative method (Questionnaire) where the respondents were from among project managers in commercial organizations that develop in-house software and organizations that develop software for external clients in the U.S.A and Australia. After analyzed the

questionnaire, the researchers found that, the following factors lead to good requirements:

1. The importance of committed and involved stakeholders;
2. Customers/ users with a high level of confidence in the development team.
3. adequate time being made available by the customers/user for requirements gathering;
4. Customers/users with realistic expectations;
5. A project managers (PM) with full authority to manage the project;
6. A PM who is above average;
7. A PM with a clear vision of the project;
8. A PM who communicates well with staff; and
9. A PM who identifies risks at the start of the project.

Another study conducted on the large companies operating in the field of the automation industry was carried out by Välimäki and Kääriäinen (2007). They focused on the best practices to support distributed business requirements management during the early phase of product development. The data collection used was mixed method through the use of questionnaire and interview. This study highlights the importance of effective and systematic management of requirement during early phases of product development.

Zainol and Mansoor (2008) reported in their study that Malaysian software industries is lacking in employing RM good practices and concluded that RM practices should be

promoted frequently during software development to enhance the quality and productivity of software product. This study was carried out on randomly chosen companies and the main instrument for data collection was questionnaire. This study has identified several issues pertaining to requirements management, such as (1) the software industry is not having a proper approach for managing requirements, (2) the companies were not aware that during their requirements management activities they are actually following the activities in CMMI model, and (3) there is lack of using best requirement management practices among the software engineers in Malaysia.

The practices of the RM and RE are very important for the development the projects as stated by many. One of this study was conducted by the Solemon, Sahibuddin and Ghani (2010) where they attempted to identify patterns of the practices of the RE for some software development companies in Malaysia. Self-administered questionnaires distributed to project managers and software developers who are working at software development companies. They found that, the overall adoption of the practices in RE in these companies is strong. Nevertheless, their study also indicted that, fewer companies use appropriate software or tools to support their requirements engineering practices.

The requirement management plays an important role when it comes to support of product development teams in the automotive industry, as stated by Gülke, Rumpe, Jansen and Axmann (2012). Furthermore Gülke *et al.* (2012) stated that, introducing or changing requirements does not only impact the product and its parts, but may lead to overhead costs in the OEM (Original Equipment Manufacturer) due to increased

complexity. Therefore, Gülke *et al.* (2012) carried out study about Automotive OEMs. In their study case study method was used as a qualitative method for collecting data, more specifically documentations.

The results from their study were, when the company introduction of new requirements or changes the existing requirements leads to three different types of costs: (1) Investment Costs, (2) Direct Costs, and (3) Overhead/ Indirect Costs. As well as, the results from case study also found that, the automotive world is getting more complex every day with a widened portfolio in brands and products and more detailed markets being all deeply connected, it needs the ideas and concepts traceability and requirements management provide. In addition, this study suggested that, more research is needed on how requirements and costs play together.

Yu and Geoffrey, (2013) conducted a study of requirements management practices for construction companies in Hong Kong, where date collection is done through semi-structure interviews and case study. The case study in this study involved the analysis of the development processes adopted by the project stakeholders that developed, designed and constructed the facilities. The finding shows that the processes and limitations of current RM practices included lack of practical framework, misinterpretation of requirements, difficulties in identifying requirements, conflicts between expectation and constraints, complex hierarchy of client's organization and communication problems in eliciting client requirements.

Requirements elicitation is one of the most critical and complex collaborative tasks in software development. Therefore, Shuhud, Richter and Ahmad (2013) in their study concentrate on this element from RM and RE. More specifically, this study examines the practices of requirements elicitation activities and understanding of how different stakeholders collaborate in the requirements elicitation process. The secondary data (prior studies) and the primary data (Interview) were the chief methods for collected data for this study. Building on their outcome, they found that the activities should be shared among other stakeholders to establish a common understanding of the requirements and that social software has the potential to support this.

In Thailand, a study was carried out by Khankaew and Riddle in 2014, to investigating the current state of requirements engineering problems and practice amongst small and medium software companies in Thailand. They conducted semi-structured interviews as the instrument for collecting data from the participants. This qualitative approach was carried out on five small companies and six medium companies.

The findings of this study indicate that SMEs in Thailand considered requirements engineering practice essential to improve development process. The four main processes consist of elicitation, analysis and negotiation, validation, and management well practiced by interviewed companies. However, the results also found, most commonly used tool among these companies was Microsoft Excel, which was used by fifty percent of the eleven companies. Although there are many requirements management tools encourage managing effectively requirements, Microsoft Office is still widely used in software enterprises in Thailand. As well, the results from this study also show that, some medium sized companies do attempt to use appropriate tool support for

requirement management activities. Such as these tools, IBM Rational (DOORS) to manage their requirements or open source requirements management tools such as the Mike tool. To give the clear picture for the prior studies, the researcher in this research summarized all the studies aforementioned above in the Table 2.7 below:

Table 2-7: Summaries the prior studies related to Requirements management in general and the practices in specific

Author (s)	Year	Objective	Method (s)	Outcome
Anderson and Felici	2001	To investigate live industrial contexts to devise product oriented approaches supporting Requirements Evolution.	Mixed Methods (Qualitative and Quantitative)	Identified an empirical framework for the analysis of Requirements Evolution
Prikladnicki, Audy and Evaristo	2003	Toanalyze the RM practices in geographically distributed environments, identifying the main challenges.	Qualitative Method (Case study)	The outcome from this study was, necessity to adopt the requirements management to distributed software development environment.
Cuevas, Serrano and Serrano	2004	To provide an accurate picture of the organization's RM processes by the use of an assessment methodology based on a two-stage questionnaire.	Quantitative Method (Questionnaire)	This study is aware that, the identification of the practices that need to be implemented is only the first step of a continuous process.
Verner, Bleistein, Cerpa and Cox	2006	To recognize what RE practices are actually used and which of these practices lead to good requirements.	Quantitative method (Questionnaires)	The results from them study were suggesting project managers vision, and communication with team members are more important than any particular background, or requirement engineering methodology they may use.

Välimäki and Kääriäinen	2007	To identify best practices to support distributed business requirements management during the early phase of product development	Mixed Methods (Qualitative and Quantitative)	This study contributing to found the important for more effective and systematic management of requirement during early phases of product development.
Zainol and Mansoor	2008	To gain an insight into the extent to which the requirements management practices have been adopted by the organizations in Malaysia	Quantitative method (questionnaire)	This study shows that, the software industry is lacking of employing good practices in managing requirements.
Solemon, Sahibuddin and Ghani	2010	To identify patterns of the practices of the requirements engineering for some software development companies in Malaysia.	Quantitative Method (Questionnaires)	The overall adoption of the practices in the requirements engineering in these companies is strong. Nevertheless, their study also indicted that, fewer companies use appropriate software or tools to support their requirements engineering.
Gülke, Rumpe, Jansen and Axmann	2012	To understanding the relationship between the requirements management and costs in Automotive Development Projects	Qualitative Method (Documentations)	The result shows that new requirements or changes on the existing requirements lead to three different types of costs: (1) Investment Costs, (2) Direct Costs,

				and (3) Overhead/ Indirect Costs.
Yu and Geoffrey	2013	To focus on the practices of projects constructed under traditional procurement system.	Qualitative method (Interviews)	They identified that, the processes and limitations of current practice included lack of practical framework, misinterpretation of requirements, difficulties in identifying requirements, conflicts between expectation and constraints, complex hierarchy of client's organization and communication problems in eliciting client requirements.
Shuhud, Richter and Ahmad	2013	To examine the practices of requirements elicitation activities and understanding of how different stakeholders collaborate in the requirements elicitation process	Qualitative Method (Documentation and Case study)	This study found, the activities should be shared among other stakeholders to establish a common understanding of the requirements and that social software has the potential to support this.
Khankaew and Riddle	2014	To investigate the current state of RE problems and practices amongst small and medium software companies in Thailand	Qualitative Method (Interview)	The results show that, the companies in Thailand encounter common problems such as, clarity, correctness, competiveness, change management and customer communication.

2.5.1 Previous Studies of Requirements Management in Malaysia

Several studies have focused to understand the RE for Malaysian software industries, especially RM. Zainol and Mansoor (2008) reported in their study that Malaysian software industries is lacking in employing RM good practices and concluded that RM practices should be promoted frequently during software development to enhance the quality and productivity of software product. This study was carried out on randomly chosen companies and the main instrument for data collection was questionnaire. This study has identified several issues pertaining to RM, such as (1) the software industry is not having a proper approach for managing requirements, (2) the companies were not aware that during their requirements management activities they are actually following the activities in CMMI model, and (3) there is lack of using best requirement management practices among the software engineers in Malaysia.

Requirements problems are widely acknowledged to reduce the quality of software and it is estimated that correcting the requirements late can cost up to 200 times as much as correcting the errors during the requirements phase. Many software projects have failed because they contained a poor set of requirements. Thus, Solemon, Sahibuddin, Ghani, & Azim (2008) studied software project problems as well as RE problems of some software companies in Malaysia. The outcome of study refers that the most of the companies experience late delivery of product problems. More than half of the companies experience budget over-runs problems. Furthermore, this study suggests that companies with CMMI-DEV certification and without any certification show no significant difference in almost all of these problems. Moreover, most of the

requirements problems experienced in the companies in the study of Solemon, Sahibuddin, Ghani, & Azim (2008) were requirements-based requirements problems rather than organizational based. However, the survey used was unable to ascertain the root cause of the problems experienced by the companies.

The practices of the RM and RE are very important for the development the projects as stated by many. One of these studies was conducted by the Solemon, Sahibuddin and Ghani (2010) where they attempted to identify patterns of the practices of the RE for some software development companies in Malaysia. Self-administered questionnaires distributed to project managers and software developers who are working at software development companies. They found that, the overall adoption of the practices in RE in these companies is strong. Nevertheless, their study also indicted that, fewer companies use appropriate software or tools to support their RE practices.

On other study, Solomon, Sahibuddin, & Ghani (2010) refers that there was no research to study both the state of the RE problems and RE practices in the Malaysia. In their study, they aims to investigate the patterns of current RE problems and practices amongst these software development companies in Malaysia through designed and conducted a survey on them. The survey questionnaires were mailed to 500 randomly selected samples of software development companies in Malaysia. The results showed that the overall adoption of the RE practices in these companies are strong. However, the results also indicated that fewer companies in the survey

have use appropriate CASE tools or software to support their RE process and practices, define traceability policies and maintain traceability manual in their projects.

In another study, Rahman, Haron, Sahibuddin, & Harun (2014) confirmed that the software projects can succeed or failed at any time during project life cycle because of poor requirement gathering and managing process. Thus, in real project development environment, some issues will appear such as miscommunication and conflict with the developers. One of the key problems of information systems requirements process is the gap between analysts and stakeholders. The objective of this study is to get the implicit and explicit experience of RE practices which were implemented during the software project requirement. Furthermore, to investigate RE issues facing the stakeholder perspectives. Based on that Rahman, Haron, Sahibuddin, & Harun (2014) there are five RE issues and challenges from the stakeholder perspectives, which is miscommunication with developer, misunderstanding during agreement process, misalignment of requirement with the business process, conflict with manager, and conflict with developer. Therefore, misunderstanding requirements are likely a risk to deliver inadequate solutions. The identified challenges include implementation of new requirements which may cause unpredictable interaction with existing requirements, requirements that are not traceable, and that requirements are too vague to be tested. The empirical study on the requirements engineering practice is conducted based on survey performed in Malaysia Public Sector.

2.6 Summary

In this chapter, more explanations and details requirements engineering is presented, moreover, requirements management in terms of the objectives and activities. The CMMI level 2 is significant to this study in order to investigate the best way to use the requirements management. As well as, this chapter sheds light the popular studies related to these phenomena.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

This chapter focuses on the research methodology used in this study. According to Mingers, (2001) the research methodology can be defined as "*structured set of guidelines or activities to assist in generating a valid and reliable research results*". It is a way of achieving the research objectives. Thus, this chapter is a presentation of the research design and the stages of methodology for the current study. It contains an overview of the research study; a discussion of the population, a description of the instruments selected for the data gathering, and outlines the strategies for data analysis and the way for interpretation.

3.2 Research Design

Research design provides plan and procedures for research that span the decisions from broad assumptions to detailed methods of data collection and analysis (Creswell, 2012). In fact there are three main approaches to data collection and analysis of the data: qualitative, quantitative, and mix method. As a philosophical underpinning for mixed methods studies, Patton (2002) convey the importance for focusing attention on the research problem in social science research and then using pluralistic approaches to derive knowledge about the problem. This study attempts to investigate the RM practices during software development projects, moreover, this study investigates using CMMI Level 2 appraisal method to identify room for improvement of RM practices of software development team in the UUM IT through

using questionnaire technique. On other hand, the managers of UUM IT will be assessed through to use of interview method. Therefore, a mixed method approach has been undertaken for achievement the objectives of study.

3.3 Stages of Research Methodology

The research methodology in this study was designed to achieve a better understanding of RM of the software development team at UUM IT. The methodology is adapted from Offermann, Levina, Schönherr and Bub (2009), which are divided into three phases problem identification, solution design, and evaluation, as described in Figure 3.1.

1 The problem identification - has been determined based on previous studies.

2 The solution design - the quantitative approach has been adopted to solve the problem through designing of questionnaire.

3 Evaluation - compare the results obtained from this study with Zainol & Mansoor, (2008), as well as, interview with the managers of UUM IT team to assessment the result.

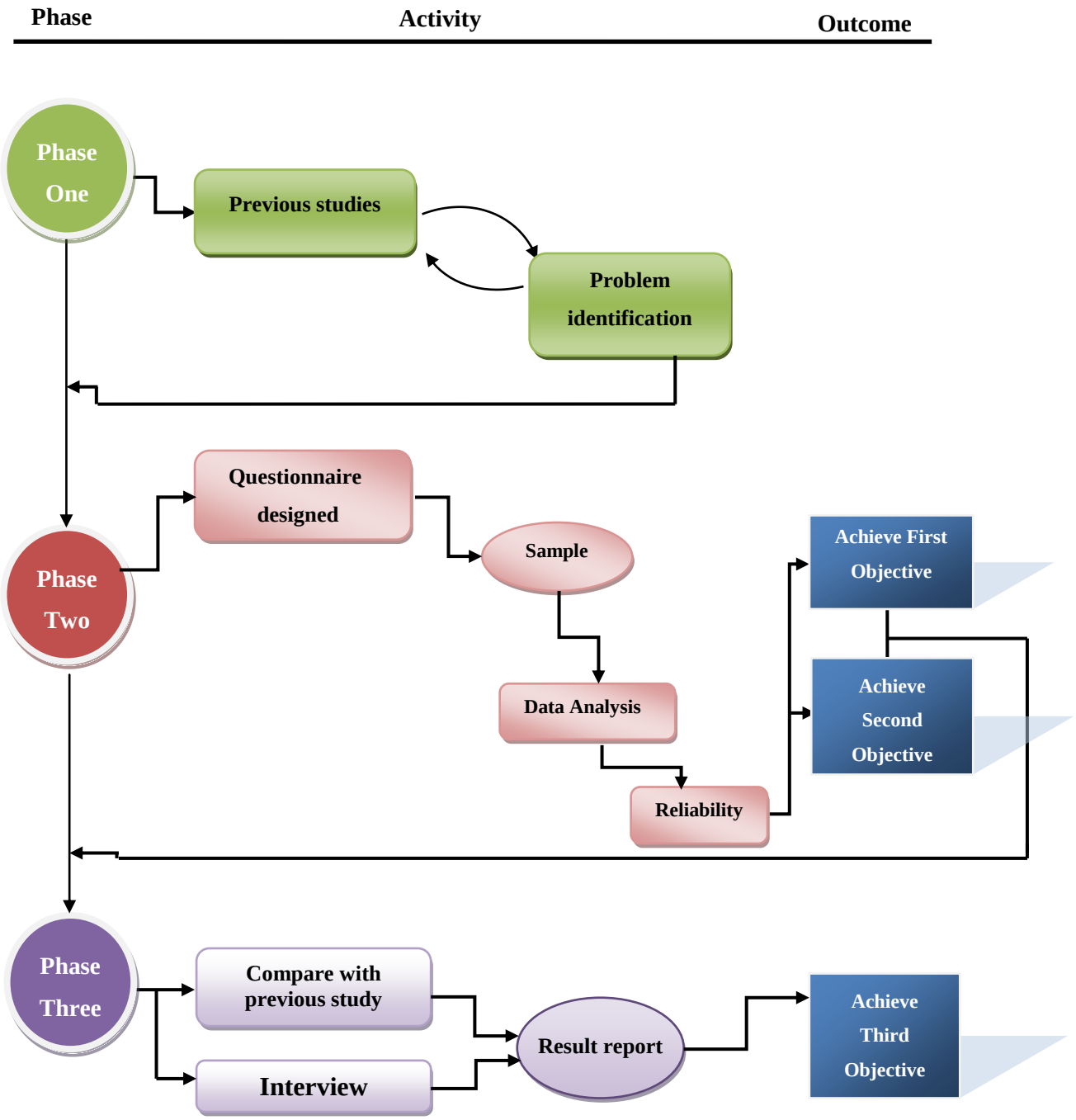

Figure 3.1: Research Framework Process

3.3.1 First Stage

3.3.1.1 Previous studies

The literature research identifies the research problem. A review on the concept of managing requirements was conducted. The RM practices was described and analyzed. Furthermore, the RM practices in CMMI Level 2 were highlighted as the one of the approaches that can be used to manage the software development process. The reviews on related work in Chapter Two has strengthened the need to propose a solution and a proper use of managing requirements.

3.3.1.2 Problem Identification

The problem criteria are important to identify the research gaps (Macintosh, Coleman, & Schneeberger, 2009). This study identifies issues in managing requirements become a global issue that indicates the cancellation of software or failure. In the context of Malaysian software industry market, this issue remains unknown with very limited studies that address it; therefore, the practical gap was discussed and determined in detail in Chapter One.

3.3.2 Second stage

The second stage comprises the following steps: the questionnaire designed, sample, data analysis, and the last step is determining reliability of the questionnaire designed. The outcome of this stage will achieve the research objective one and research objective two.

3.3.2.1 Questionnaire designed

Questionnaire can be defined as a set of questions that are answered by the respondents, whose responses are recorded (Sekaran & Bougie, 2010). The questionnaires were prepared based on Felicia (2001), Zainol and Mansoor (2008), Anderson & Felici (2001), Schulze & Pretorius (2013), Weber, Curtis & Chrissis (1995) and Khankaew & Riddle (2014) which was prepared and divided into four sections. The section A is identified as the demographic profile for the responders. The background variables used in this study are position of participation, experience, IT professionals, and participation in the software system projects.

The section B relates with the RM processes. The section C describes the items to achieve the RM activities. The items for section D are focused on the RM activities in CMMI Level 2. In order to ensure that the questionnaire is applicable to achieve the research objectives, Table 3.1 illustrates the matching between research objectives, the sections of questionnaire and the sources of the adopted questionnaires.

Table 3-1: Questionnaire items to achieve the study objectives

Research objective	Questions	References
Achieve First Objective	1. Do you have standards templates/documents for describing requirements? 2. Do you have guidelines how to write requirements? 3. Do you use UML diagram to document all the requirements? 4. Do you supplement natural language with other descriptions of requirements? 5. Do you specify requirements with unique identification?	(Anderson & Felici, 2001; Zainol & Mansoor, 2008). Schulze & Pretorius, 2013)
Achieve Second Objective	1. Do you carry out a feasibility study before starting a new project? 2. Do you develop a prototype in order to understand poor or complex requirements? 3. Do you reuse requirements from other systems which have been developed in the same domain? 4. Do you define system boundaries? 5. Do you develop a checklist for requirement analysis? 6. Do you list all your requirements in requirement analysis? 7. Do you priorities requirements? 8. Do you perform any risk analysis on requirements? 9. Do you use prototyping to demonstrate requirements for validation? 10. Do you uniquely identify each requirement? 11. Do you have defined policies for requirements management? 12. Do you use a database to manage requirements? 13. Do you identify volatile requirements? 14. Do you record rejected requirements? 15. Do you reuse requirements over different projects?	(Weber, Curtis & Chrissis, 1995; Anderson & Felici, 2001; Zainol & Mansoor, 2008). Khankaew & Riddle, 2014)

Achieve Third Objective	*Activity 1: The software engineering group reviews the system requirements before they are incorporated into the software project.* .	
	1. Incomplete and missing system requirements are identified.	(Chrissis & Weber, 1993; Zainol & Mansoor, 2008)
	2. The system requirements are reviewed to determine they are feasible and appropriate to implement, clearly stated, consistent and testable.	
	3. Any system requirement identified as having potential problems are reviewed and the necessary changes are made	
	4. Commitments resulting from the system requirements are negotiable with the affected groups (eg:software engineering, software estimating, system test, ect)	
	Activity 2: The software engineering group uses the system requirements as the basis for software plans, work products and activities.	
	1. The system requirements are managed and controlled	(Chrissis & Weber, 1993; Zainol & Mansoor, 2008)
	2. The system requirements are the basis for the software development plan.	
	3. System requirement are the basis for developing the software requirements.	
	Activity 3: Changes to system requirements are reviewed and incorporated into the software project.	
	1. The impact to existing commitments is assessed, and changes are negotiable as appropriate.	(Chrissis & Weber, 1993; Zainol & Mansoor, 2008)
	2. Changes that need to be made to the software plans, work products, and activities resulting from changes to the system requirements are identified, evaluated, assessed to risk, documented, planned, communicated to the affected groups and individuals and tracked to completion.	

3.3.2.2 Sampling

Sampling is the process of selecting units (e.g., organizations, people) from a population of interest so that by studying the sample we may fairly generalize the results back to the population from which they were chosen (Sekaran & Bougie, 2010) .Target sample for this study is for UUM IT.

The sample size of the UUM IT professionals' is 24 employees; they are involved in the software development processes. The questionnaire was distributed to all the IT professionals' employees in UUM IT to ensure the reliability and validity of the results. Based on the findings by Sekaran and Bougie (2010), the present study has identified a sample size of 23 respondents who meet the population inclusion criteria set forth in this study.

3.3.2.3 Data Analysis

Upon completion of data collection, the data was analyzed using descriptive statistics with frequencies procedure technique. This procedure gives frequency tables. It is usually used for categorical data. This statistics method allows researcher to choose descriptive measures such as percentiles. Specifically, SPSS version 19 was used for data analysis. Furthermore, the confidentially, privacy, preservation from distortion, and anonymity, shall be guaranteed in this study.

3.3.3 Third stage

3.3.3.1 Compare with previous study

The major aim of comparative research is to identify similarities and differences between social entities. Comparative research, simply put, is the act of comparing two or more things with a view to discovering something about one or all of the things being compared (Rose, 2004). The majority agreement is that there is no methodology peculiar to comparative research (Heidenheimer, Heclo & Adams, 1983). This study focuses on using of CMMI level 2 in the managing requirement of software development in the UUM IT. The result of this study is compared with the previous study for Zainol and Mansoor (2008), which addressed the used of CMMI requirement management process area to develop the software engineering in Malaysia.

3.3.3.2 Interview

Many methods are acceptable to evaluate the result of studies; one of these methods in qualitative approach is interview the experts to assess the outcome of studies (Creswell, 2012). Semi-structured interviews process is selected to manage this interview. Semi-structured interviews are based on the use of an interview guide, with an open-end question. According to Patton (2002), the sample size in qualitative inquiry depends on the cases, which means there is not rules for sample size. In this study, five of experts from UUM IT managers will be participant in the interview.

3.4 Summary

This chapter described the research methodology used in the present study to discuss the methods to collect data, which includes the questionnaire that was selected from previous study. Finally, this chapter described the methods of data analysis aimed at answering the research questions of this study. The succeeding chapter will discuss and analyze the findings of the current study.

CHAPTER FOUR
THE RESULTS

This chapter presents results of the survey which was conducted at the UUM IT. Thereafter, gap analysis was carried out to identify strengths, weaknesses and improvement areas. SPSS was used to analyze the data. The report on these sections is based on the data provided by 23 respondents of information technology experts within UUM IT. The chapter is divided into five sections. The first section illustrated the overview for the UUM IT. The second section discusses the response rate and the descriptive statistics of the respondents' demographic profiles. The third section explains the requirements description. The fourth section presents the descriptive statistics of RM practices. The final section details a RM practices to identify in CMMI level 2.

4.1 UUM Information Technology

UUM IT is located in UUM main campus, Sintok with facilities and support staff distributed all over campus. Core computing and network services at UUM are provided by UUM IT, which ensures safe and secure access to enterprise systems and the campus network. This center delivers services that support research, learning, teaching, administration and student life. All IT services are supported by a front-line IT help desk and IT zone support staff. This organization aims to make every members of the UUM community have access to his/her data and everyday resources from anywhere, backed by the technology and support his/her need. UUM

IT is made up of nearly 150 employees working together in teams across several divisions.

4.2 Demographic Profiles

This section describes the demographic profiles of the respondents who participated in the study. Prior to reporting the main findings of the survey, the demographic profiles of the respondents must be identified. The detection of out-of-range values can be achieved using descriptive analysis and the frequency method (Dillon, Madden, & Firtle, 1990). Demographic profiles include participants' position, experience, IT professionals, and participate in the software system development.

Table 4.1 indicates that majority of the respondents are programmers (65.2%). The other respondents were divided among software developers (21.7%); and IT professionals (8.7%). As regards the experience of the respondents, most of the respondents have experience among 6 to 10 years (37.8%), followed by the group of more than 20 years experience (26.1%), then 1 to 5 years experience (21.7%), and 11 to 19 years experience (17.4%). This observation indicates the dominance of experienced, employees in UUM IT developer team.

In the terms of IT professionals, results show that the highest number of respondents refers there are over 50 professionals (69.6%), while only 4.3% of respondents refers there are less than 10 IT professionals. Thus, majority of the respondents believed that the UUM IT has considerable professionals of IT. In the terms of participation in

the software system projects, 10 respondents (43.5%) participated in 6-10 software system projects of software development since they started working, while 7 respondents (30.4%) had participated in less than 5 software systems projects previously. Whereas, 4 of respondents (17.4%) have participated between 11 to 19 projects of software systems development, followed by the 2 of respondents (8.7%) have participated in more than 20 software systems projects.

Table 4-1: Descriptive statistics of respondents' demographic profiles

Demographic	Characteristics	Frequency	Percent%
Participants' position	Software developer	5	21.7%
	IT professional	2	8.7%
	Programmer	15	65.2%
	Other	1	4.3%
Total		23	100.0
Experience	1 to 5 years	5	21.7%
	6 to 10 years	8	34.8%
	11 to 19 years	4	17.4%
	more than 20 years	6	26.1%
Total		23	100.0
IT professionals	Less than 10 professionals	1	4.3%
	20-49 professionals	6	26.1%
	Over 50 professionals	16	69.6%
Total		23	100.0
Participate in the software system projects	1 to 5 software systems	7	30.4%
	6 to 10 software systems	10	43.5%
	11 to 19 software systems	4	17.4%
	more than 20 software systems	2	8.7%
Total		23	100.0

The result was observed in the UUM IT that the majority of professionals working have experience more than 5 years, where Kolb (2014) refers that the experience of

employees can be a source of learning and development. On other side, even the IT professional in UUM IT team are limited, the percentage of programmer is high. This is expected because it is generally observed in the organization that the majority of professionals working in the field of IT are programmers. Furthermore, improvement of performance of employees have effect on the level of acceptance for a new approach (Zainol & Mansoor, 2008). Therefore, the UUM IT team participation in the software system projects with more than 5 projects is near to be 70%, which means that most likely the longer years of working experience would improve the performance of employees. These performance improvements have a positive effect on accepting the new working approaches (Zainol & Mansoor, 2008). Thus, the possibility of adopting an approach to the development work in software projects is acceptable.

4.3 Reliability Analysis

According to Sekaran (2003), the closer the reliability coefficient gets to 1.0, the better it is, and those values over .80 are considered as good. Those value in the .70 is considered as acceptable and those reliability value less than .60 is considered to be poor (Sekaran, 2003). According to George and Mallery (2003), reliability is referred to as the degree to which a measure is free from error and therefore yields consistent results.

Table 4-2: Reliability analysis

Variables	No. of Items	Cronbach'sAlpha
Requirement management description	5	.785
Requirement management practices	15	.777
Activity one of CMMI	4	.991
Activity two of CMMI	3	.985
Activity three of CMMI	2	.991

As shown in Table 4.2, alpha value for independent variable and dependent variables is about and above 0.60 which is considered as accepted value.

4.4 Requirements Management Description

RE processes have several stages (Sommerville, 2009), one of the important stage is RM, especially, when software system becomes increasingly larger (Zainol & Mansoor, 2008). IT professional refers to as a person who has technical knowledge and skills in ICT (Joseph, Ang, Chang & Slaughter, 2010). IT professional in organizations today work within a complex landscape of interconnections and interdependencies, and much of their work requires the skillful orchestration and management of those ties (Gallagher, Gallagher & Kaiser, 2013). This section focuses on the description of the RM practices for the UUM IT developer team. According to Anderson and Felicia (2001), Zainol and Mansoor (2008), gives five questions to identify the RM practices. Table 4.2 illustrated the result of the second section of the questioners.

Table 4-3: Descriptive statistics of requirements management practices

	Requirements Description	Characteristics	Frequency	Percent%
1	Do you have standards templates/documents for describing requirements?	Never	2	8.7%
		Rarely	0	0%
		Sometimes	6	26.1%
		Regularly	8	34.8%
		Always	7	30.4%
	Total		23	100.0
2	Do you have guidelines how to write requirements?	Never	3	13.0%
		Rarely	0	0%
		Sometimes	4	17.4%
		Regularly	11	47.8%
		Always	5	21.7%
	Total		23	100.0
3	Do you use UML diagram to document all the requirements?	Never	1	4.3%
		Rarely	3	13.0%
		Sometimes	15	65.2%
		Regularly	3	13.0%
		Always	1	4.3%
	Total		23	100.0
4	Do you supplement natural language with other descriptions of requirements?	Never	1	4.3%
		Rarely	2	8.7%
		Sometimes	8	34.8%
		Regularly	10	43.5%
		Always	2	8.7%
	Total		23	100.0
5	Do you specify requirements with unique identification?	Never	4	17.4%
		Rarely	4	17.4%
		Sometimes	12	52.2%
		Regularly	2	8.7%
		Always	1	4.3%
	Total		23	100.0

The output in Table 4.2 shows that the developer team of UUM IT mostly uses standards templates /documents to determine the requirements for their projects.

Where 7 respondents (30.4%) are used always, then 8 respondents (34.8%) are regularly to use standards templates /documents, while only 2 respondents (8.7%) are never used it. This observation indicates that the developer team of UUM IT followed the rule to use standards templates /documents to identify the requirements for projects. Moreover, this team uses the guidelines to write requirements in the templates /documents, where 11 respondents (47.8%) are regularly to use guidelines and 5 respondents (21.7%) uses guidelines always. While 3 respondents (13%) indifferent to use guidelines for write the requirements of projects.

In the terms of using UML diagram, results show that the highest number of respondents were 15 (65.2%) are sometimes using UML diagram to document the requirements, whereas only one respondent (4.3%) never uses UML diagram. On the other hand, 10 respondents (43.5%) combine the natural language with descriptions of requirements, whereas only one respondent (4.3%) never uses natural language. The developer team of UUM IT sometimes specifies their requirements with unique identification with 12 respondents (52.2%). Figure 4.1 gives an representation of overview diagram for the RM practices of UUM IT.

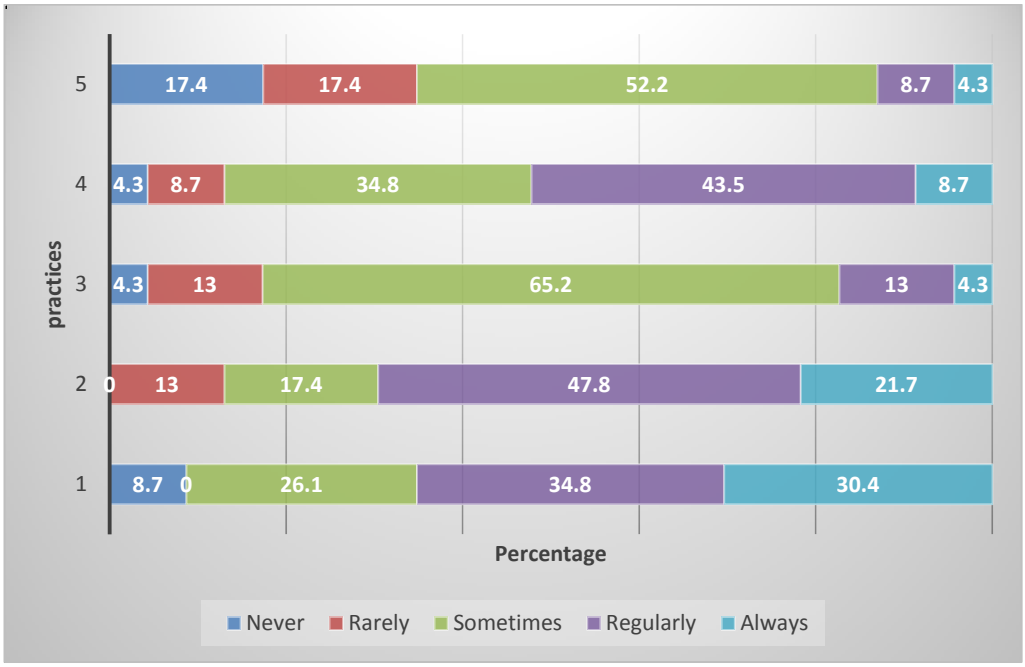

Figure 4.1: Requirements descriptive practices diagram

The result has revealed that the UUM IT team sometimes uses UML diagrams to document their requirements, which can be a powerful method for understanding the behavior of the whole system (Purbasari, Iping, Santoso & Mandala, 2013). In addition, UUM IT has standard templates/document and guidelines on how to write requirements, where the percentage that always practice both of these practices is high with 65%. In contrast, the percentage to use the supplement natural language with other descriptions of requirements and specify requirements with other unique identification is more than 50%, which refers that the developer team in UUM IT usually uses the RM description tools to identify the requirements of software projects.

4.5 Requirement Management Practices

RM is one of the RE activities that focuses on managing requirements over the entire software development. This section focuses on investigating the activities for the RM practices within software UUM IT through reviewing 15 activities based on Paulk, Curtis, Chrisis and Weber, (1993); Paulk, Weber, Curtis and Chrisis, (1995); Zainol and Mansoor (2008). The result of the survey was illustrated in Table 4.3

The output of the questioner indicates that majority of the respondents confirmed that they use feasibility study before they begin the implementation of any project, where 10 respondents (43.5%) are always use it and 9 respondents (39.1%) are regularly. On other hand, most of respondents, 10 (43.5%) regularly, 4 (17.4%) always, are develop prototype for identify the requirements, specially the poor or complex one. Furthermore, the IT employees in the UUM IT take advantage of the systems that have been developed previously to identify new projects requirements, where 7 respondents (30.4%) are regularly with this activity and 4 respondents (17.4%) are always.

Table 4-4: Descriptive statistics of requirements management activities

	Requirements management Practices	Characteristics	Frequency	Percent%
1	Do you carry out a feasibility study before starting a new project?	Never	0	0%
		Rarely	3	13%
		Sometimes	1	4.3%
		Regularly	9	39.1%
		Always	10	43.5%
	Total		23	100.0
2	Do you develop a prototype in order to understand poor or complex requirements?	Never	1	4.3%
		Rarely	2	8.7%
		Sometimes	6	26.1%
		Regularly	10	43.5%
		Always	4	17.4%
	Total		23	100.0
3	Do you reuse requirements from other systems which have been developed in the same domain?	Never	1	4.3%
		Rarely	3	13.0%
		Sometimes	8	34.8%
		Regularly	7	30.4%
		Always	4	17.4%
	Total		23	100.0
4	Do you define system boundaries?	Never	0	0%
		Rarely	3	13.0%
		Sometimes	7	30.4%
		Regularly	8	34.8%
		Always	5	21.7%
	Total		23	100.0
5	Do you develop a checklist for requirement analysis?	Never	1	4.3%
		Rarely	3	13.0%
		Sometimes	7	30.4%
		Regularly	8	34.8%
		Always	4	17.4%
	Total		23	100.0
6	Do you list all your requirements in requirement analysis?	Never	0	0%
		Rarely	1	4.3%
		Sometimes	10	43.5%
		Regularly	8	34.8%
		Always	4	17.4%
	Total		23	100.0
7	Do you priorities requirements?	Never	0	0%
		Rarely	1	4.3%

		Sometimes	2	8.7%
		Regularly	13	56.5%
		Always	7	30.4%
	Total		23	100.0
8	Do you perform any risk analysis on requirements?	Never	1	4.3%
		Rarely	4	17.4%
		Sometimes	7	34.8%
		Regularly	10	43.5%
		Always	1	4.3%
	Total		23	100.0
9	Do you use prototyping to demonstrate requirements for validation?	Never	1	4.3%
		Rarely	5	21.7%
		Sometimes	6	26.1%
		Regularly	9	39.1%
		Always	3	13.0%
	Total		23	100.0
10	Do you uniquely identify each requirement?	Never	1	4.3%
		Rarely	2	8.7%
		Sometimes	6	26.1%
		Regularly	10	43.5%
		Always	4	17.4%
	Total		23	100.0
11	Do you have defined policies for requirements management?	Never	2	8.7%
		Rarely	0	0%
		Sometimes	6	26.1%
		Regularly	14	60.9%
		Always	1	4.3%
	Total		23	100.0
12	Do you use a database to manage requirements?	Never	8	34.8%
		Rarely	5	21.7%
		Sometimes	3	13.0%
		Regularly	6	26.1%
		Always	1	4.3%
	Total		23	100.0
13	Do you identify volatile requirements?	Never	4	17.4%
		Rarely	5	21.7%
		Sometimes	6	26.1%
		Regularly	6	26.1%
		Always	2	8.7%
	Total		23	100.0
14		Never	2	8.7%

	Do you record rejected requirements?	Rarely	8	34.8%
		Sometimes	9	39.1%
		Regularly	4	17.4%
		Always	0	0%
	Total		23	100.0
15	Do you reuse requirements over different projects?	Never	1	4.3%
		Rarely	1	4.3%
		Sometimes	10	43.5%
		Regularly	8	34.8%
		Always	3	4.3%
	Total		23	100.0

In terms of identifying system boundaries, results show that the highest number of respondents was 8 (34.8%) are regularly determined the system boundaries and 5 respondents (21.7%) are always, which means most of the employees identify boundaries of the system that they developed. Moreover, half of respondents 8 (34.8%) regularly, 4 (17.4%) always, are agree with using checklist for requirement analysis, when they begin the implementation of any project. Also, half of respondents 8 (34.8%) regularly, 4 (17.4%) always are using list to identify their requirements in the requirement analysis. But, most of respondents 13 (56.5%) regularly, 7 (30.4%) always, have arranged the priorities of their requirements.

Risk analysis on requirements is one of important activities for the respondents, where 7 (30.4%) sometimes, 10 (43.5%) regularly, 1 (4.3%) always, followed the rules of risk analysis when they develop any system. Therefore, most of respondents are validate their requirements through use of prototyping, where 6 (26.1%) sometimes, 9 (39.1%) regularly, 3 (13%) always. On other hand, the respondents preferred to identify each requirement for their projects uniquely, 10 (43.5%)

regularly, 4 (17.4%) always. Furthermore, most of the respondents 14 (60.9%) regularly, 1 (4.3%) always, focus on defining policies for requirements management in the projects.

The survey show that use of a database to manage requirements is not acceptable from the respondents, where 8 respondents (34.8%) said they are not using database to manage requirements, and 5 respondents (21.7%) are rarely, while only one respondent (4.3%) using database to manage requirements. Whereas, to identify volatile requirements has oscillated between respondents to use it or not. Moreover, in recording the rejected of requirements, most of the respondents 2 (8.7%) never, 8 (34.8%) rarely, 9 (39.1%) sometimes, are not record the rejected requirements. By contrast, most of respondents reuse the requirements over the different projects, where 6 (26.1%) sometimes, 9 (39.1%) regularly, 3 (13%) always. Figure 4.2 shows the level of usage of each requirement management activity from the UUM IT employees.

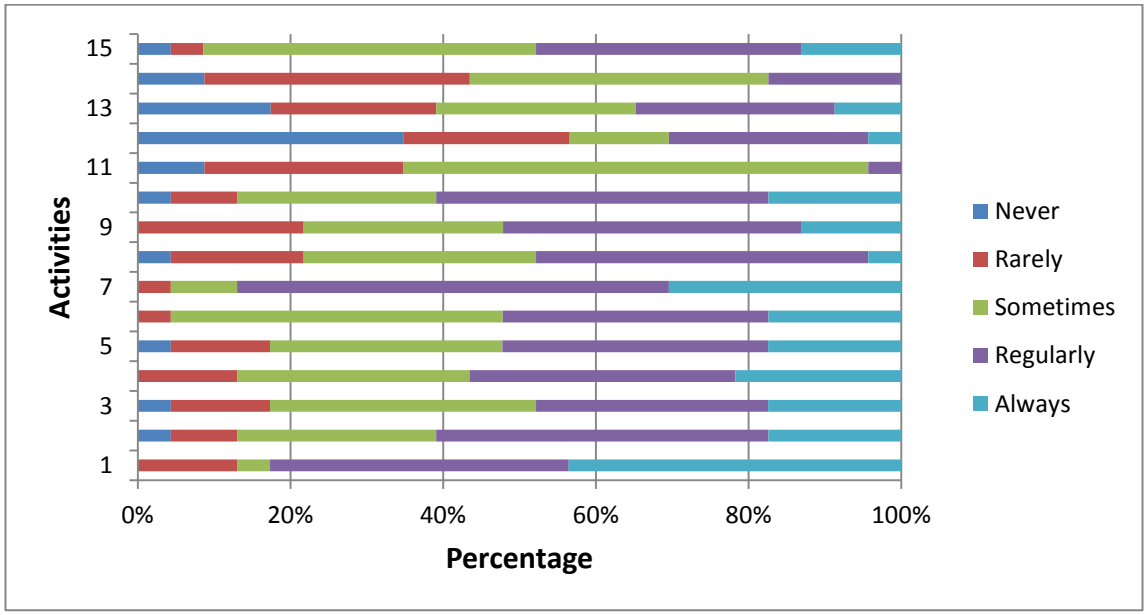

Figure 4.2: Level of usage of each requirement management activity for UUM IT

.

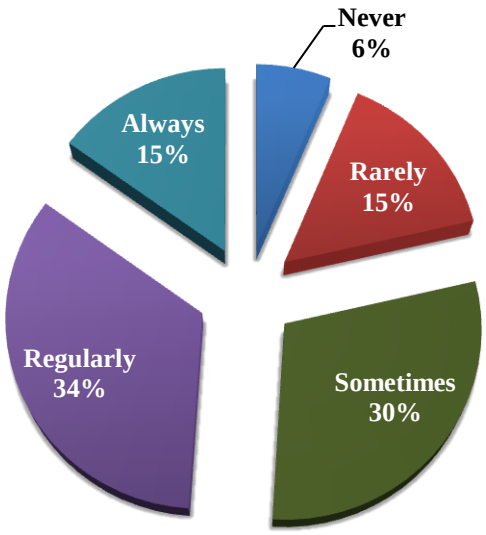

Figure 4.3: Requirement management activities for UUM IT

Based on above figure, the developer team in the UUM IT are using the RM to develop their projects with positive percentage reach 50% (Regularly and Always), even though, 21% (never and rarely) are limited or do not use the activities of RM. Even though, this may reduce the success of software projects, it is still positive that UUM IT is aware of the needs for RM activities. Thus, the CMMI model can to be need a key in achieving this aim, which is not employed in UUM IT as describe in next section.

4.6 Requirement Management in CMMI Level 2

CMMI level 2 consists of the best practices for the development and maintenance activities of products and services within the RM (Crespo & Ruiz, 2012). In this context, CMMI have been adopted to evaluate the UUM IT capabilities, within framework of RM. According to Paulk et al. (1993); Zainol and Mansoor (2008), there are three main activities for CMMI, which should be performed in the process of RM. For the first stage, the familiarity of developer team of UUM IT with the CMMI level 2 has to been assessed.

In Table 4.4, results show that the highest number of respondents 17 (73.9%) has never heard about CMMI; while 4 respondents (17.4%) have heard of the CMMI model. Even so, the CMMI has been used occasionally by only two respondents, which is an indicator that the spread of CMMI model in UUM IT is limited.

Therefore, most of the respondents cannot evaluate the CMMI model, because they have never heard about it. Based on figure 4.4, the three main activities for the

CMMI have been evaluated by only the respondents that have heard or used CMMI, which means only 6 respondents, have been dealing with CMMI. Moreover, the main three activities percentage for the CMMI activity is illustrated in Figure 4.4

Table 4-5: The level of CMMI familiarity in UUM IT

Item	Characteristics	Frequency	Percent%
Have you used or heard about Capability Maturity Model (CMM) or Capability Maturity Model Integration (CMMI)??	Never heard of it	17	73.9%
	Heard of it	4	17.4%
	Use it occasionally	2	8.7%
Total		23	100.0

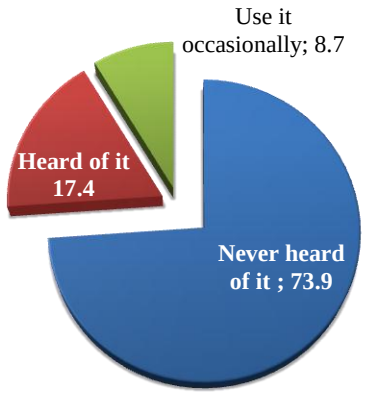

Figure 4.4: The level of CMMI Familiarity

For the first activity of CMMI, is about reviewing of the system requirements before they are incorporated into the software project. Table 4.5 describes the result for the respondents.

In terms of identifying the incomplete and missing system requirements, results show that the highest number of respondents was 5 (83.3%) are sometimes, which means to identify the system requirements errors is rarely used. On other hand, most of respondents 4 (66.6%) are regular and always, which means the system requirements had been reviewed. The potential problems for system requirement that are faced by the IT professional in the UUM IT 4 (66.6%) are regularly reviewed and the necessary changes are made. In addition, commitments resulting from the system requirements are also regular. This indicates that most of the IT professional employees in the UUM IT perform these sub-activities in their requirements management phase.

Table 4-6: Activity 1 for CMMI

Activity 1		Characteristics	Frequency	Percent%
1	Incomplete and missing system requirements are identified.	Never	0	0%
		Rarely	0	0%
		Sometimes	5	83.3%
		Regularly	1	16.7%
		Always	0	0%
	Total		6	100.0
2	The system requirements are reviewed to determine they are	Never	0	0%
		Rarely	1	16.7%
		Sometimes	1	16.7%

	feasible and appropriate to implement, clearly stated, consistent and testable.	Regularly Always	2 2	33.3% 33.3%
	Total		6	100.0
3	Any system requirement identified as having potential problems are reviewed and the necessary changes are made	Never Rarely Sometimes Regularly Always	0 0 1 4 1	0% 0% 16.7% 66.6% 16.7%
	Total		6	100.0
4	Commitments resulting from the system requirements are negotiable with the affected groups (e.g.: software engineering, software estimating, system test, etc.)	Never Rarely Sometimes Regularly Always	0 0 2 3 1	0% 0% 33.3% 50.0% 16.7%
	Total		6	100.0

The second activity of CMMI refers to the use the system requirements as the basis for software plans, work products and activities. Table 4.6 shows the result that refers the most of the developer team of the UUM IT manage and control the system requirements. Moreover, the basic of software development plan is system requirements with 2 (33.3%) regularly, and 2 (33.3%) always. On other hand, system requirements is the basis for most of respondents 4 (66.6%)

Table 4-7: Activity 2 for CMMI

Activity 2	Characteristics	Frequency	Percent%
	Never	0	0%
	Rarely	0	0%

1		Rarely	1	16.7%
	The system requirements are managed and controlled	Sometimes	2	33.3%
		Regularly	3	50.0%
	Total		6	100.0
		Never	0	0%
2	The system requirements are the basis for the software development plan.	Rarely	1	16.7%
		Sometimes	1	16.7%
		Regularly	2	33.3%
		Always	2	33.3%
	Total		6	100.0
		Never	0	0%
		Rarely	0	0%
3	System requirement are the basis for developing the software requirements.	Sometimes	2	33.3%
		Regularly	4	66.6%
		Always	0	0%
	Total		6	100.0

The third activity of CMMI is review of changes to system requirements and their incorporation to the software project. Table 4.7 shows that the impact to existing commitments is assessed, and changes are negotiable as appropriate, where 3 (50.4%) rarely, 2 (33.3%) regularly, 1 (16.7%) always. On other hand, the majority of respondents regularly (50%) to changes that need to be made to the software plan. To conclude, the uses of CMMI level 2 to change the requirements are still very limited.

Table 4-8: Activity 3 for CMMI

Activity 3		Characteristics	Frequency	Percent%
		Never	0	0%
1	The impact to existing commitments is	Rarely	3	50.0%

	assessed, and changes are negotiable as appropriate.	Sometimes	0	0%
		Regularly	2	33.3%
		Always	1	16.7%
	Total		6	100.0
2	Changes that need to be made to the software plans, work products, and activities resulting from changes to the system requirements are identified, evaluated, assessed to risk, documented, planned, communicated to the affected groups and individuals and tracked to completion.	Never	1	16.7%
		Rarely	2	33.3%
		Sometimes	0	0%
		Regularly	3	50.0%
		Always	0	0%
	Total		6	100.0

Actually, the CMMI model, which is a key in achieving RM, is not widely known among the developer team of UUM IT since the percentages of those known and using it regularly or is relatively low. It is interesting to note that familiarity with the models is high, but in practice this does not indicate that the developer team of UUM IT which know about them really appreciate their use and effectiveness.

The results of CMMI activities showed that the developer team of UUM IT is regularly conducting all the sub-activities. On the other hand, the percentage of that developer team of UUM IT who rarely or never using these sub-activities is relatively low. The result of this study indicates that the developer team of UUM IT is familiarity with CMMI model is high but in practice it is not implemented effectively.

4.7 Qualitative Approach to Measure Requirements Management Practices

Qualitative approach is a method of inquiry employed in many different academic disciplines. Creswell (2012), stated that the qualitative approach intents is not to

generalize, but to develop an in-depth exploration of a central phenomenon, which is best achieved by using purposeful sampling strategies. Thus, the data is used to collect both quantitative survey data with UUM IT team and interviews with the managers of the UUM IT. Knodel and Saengtienchai (2005) confirmed that the reflecting on the use of both forms of data to understand the problem is better because quantitative data alone would be inadequate. The current study had also a qualitative method design using semi-structured interview technique with sample of five from UUM IT managers. The interview was applied face to face with 15 questions about the RM practices. The outcomes of the interviews are illustrated in Table 4.9.

Table 4-9: Descriptive statistics of requirements management activities through managers' perspective

Requirements management Practices		Characteristics	Frequency	Percent%
1	Do you carry out a feasibility study before starting a new project?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
2	Do you develop a prototype in order to understand poor or complex requirements?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
3	Do you reuse requirements from other systems which have been developed in the same domain?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
4	Do you define system boundaries?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**

5	Do you develop a checklist for requirement analysis?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
6	Do you list all your requirements in requirement analysis?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
7	Do you priorities requirements?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
8	Do you perform any risk analysis on requirements?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
9	Do you use prototyping to demonstrate requirements for validation?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
10	Do you uniquely identify each requirement?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
11	Do you have defined policies for requirements management?	Yes	1	20%
		No	4	80%
	Total		**5**	**100.0**
12	Do you use a database to manage requirements?	Yes	0	0%
		No	5	100%
	Total		**5**	**100.0**
13	Do you identify volatile requirements?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**
14	Do you record rejected requirements?	Yes	1	20%
		No	4	80%
	Total		**5**	**100.0**
15	Do you reuse requirements over different projects?	Yes	5	100%
		No	0	0%
	Total		**5**	**100.0**

Descriptive for the result for each question are illustrated in Table 4.8, this table displays the outcome from the managers for each item. For the first item it is observed that the all those interviewed have positively answer about carrying out a feasibility study before starting a new project, where they pointed out that the UUM IT developer team discuss in detail the requirements of the project before starting the project. This answer corresponds with the results of the questionnaire for UUM IT team. Moreover, they also have positively answered about developing a prototype in order to understand poor or complex requirements. The managers pointed out that developing prototype is based on user requirements, where the UUM IT team discussed the requirements to develop prototype and sharing with customer to assess it. This result also matched with the UUM IT team answers. On other hand, for the third item it is observed that the all those interviewed have positively answer about reuse of requirements from other systems that have been developed in the same domain. They pointed out that the UUM IT developer team analyzes the requirements and if the requirements exist in other systems will they reuse it. This answer corresponds with the results of the questionnaire from the UUM IT team.

All the participants stressed the importance of the definition of system boundaries, where they pointed out that the system boundaries is defined based on size, start, and end date. Moreover, all the participants refers that the UUM IT team used checklist for requirements analysis, where they pointed out that the UUM IT team write all requirements analysis in checklist to process it one by one. As well as, the participants also stressed that the UUM IT team uses a list for all the requirements,

where they refers that the team listed the requirements after collected them from the users. Based on that, the managers of UUM IT team clarify the importance of managing the requirements through the use of checklist and list of requirements analysis to pursue work on a regular basis. For the seventh item, all the participants refers that the UUM IT team depends on the priorities requirement , where they pointed out that each requirement is given a priority based on importance.

Risk analysis on requirements is a very important activity to requirements management, where all the participants stressed on it. The participants also pointed out that the UUM IT team identify the risk and recorded the importance finding, then check out the necessary update. On the other hand, all the participants refer that the requirements have to be validated through the development prototype. Furthermore, all the participation stressed that the UUM IT team has uniquely identify for each requirement. In contrast, one of the participants refers that there is not clear policies for RM, while the four participants stressed the opposite. On the other hand, all participants confirmed that the UUM IT team do not use a database to manage requirements, where they refer that the team wrote all the requirements in electronic file and saved it.

For item thirteen, the output displays that all the participants refer that the UUM IT team identify volatile requirements if there is a necessity for mentioning them, where the team notices that volatile requirements to discuss with the client. However, for the fourteenth item, one of the participants refer that UUM IT team record rejected

requirements, while the other four participants stressed that the UUM IT team most of the time ignore recording of the rejected requirements. Moreover, all the participants stressed that the UUM IT team identifies all the requirements of a new system and check out with other projects to find similar requirements to reuse. Thus, the UUM IT team has always reused requirements from different projects.

As a result, the UUM IT managers answered the interview, their answers have confirmed the outcome of the questionnaire that have been got from UUM IT team. The results for 15 questions of the requirement management activities are similar, thus, the use of both forms of data collection method give more ability to understand the current situation for RM practices applied at UUM IT.

4.8 Compare results with Previous Study

In the previous sections, the results of this study have been presented. Hence, the results will be assessed through using comparative method with the previous study to validate the current study; also, to investigate the requirements management development in the UUM IT. Furthermore, the outcome of the current study will be compared with the previous study form Zainol and Mansoor (2008), which addressed the used of CMMI to develop the software projects in Malaysia.

Based on a study conducted for companies in Malaysia, the percentage to describe the requirements by using standard templates/documents for Malaysian companies is 40%. For our study, this percentage has been increased to reach 65% in the UUM IT,

which refers that the describing of the requirements through use standard templates/documents has been improved (Regularly and Always see Table 4.3). On other hand, Zainol and Mansoor (2008) emphasizes that the UML is not regularly used to describe requirements in the companies, this fact is confirmed by our study, while the UUM IT developer team sometimes use UML to describe requirements. Moreover, the companies are regularly described requirements with natural language, which are similar with the current study (see Table 4.3). The companies are sometimes specifying their requirements with unique identification; this situation applies to UUM IT also (see Table 4.3) In general, the outcome of the current study shows that there is a similarity with the previous study, which is a weakness point in RM development, but there is a little improvement.

Zainol and Mansoor (2008) state that less than 40% of the Malaysian companies have employed all RM activities, where this percentage has been increased to become around 50% in the UUM IT (see Table 4.4). Thus, it is an indication that there is some interest in the use of RM activities in UUM IT. In details, before starting new projects, the Malaysian companies always carry out feasibility studies, which is always carried out in the UUM IT. On other hand, manage requirements through using database in the Malaysian companies are rarely, as well as, reuse requirements over different projects. This fact has confirmed in the UUM IT, where 34% of the UUM IT developer team never use database to manage their requirements (see Table 4.4). Although, 19% Malaysian companies tend not apply requirements management or limited, but Zainol and Mansoor (2008) argues it is still

positive with four-fifths of the Malaysian companies were understand the significance of requirement management activities. This percentage has increased to become 21% in the UUM IT (see Figure 4.4) Based on above, the developer team in the UUM IT are using the requirements management to develop their projects with positive percentage reach 50% (Regularly and Always), even though, 21% (never and rarely) are limited or do not use the activities of requirements management. Even though this may reduce the success of software projects, it is still positive that UUM IT was aware of the need for RM activities. Thus, the CMMI model can to be a key in achieving this aim, which is not employed in UUM IT as describe in next section, even though we agree about the positive point for Zainol and Mansoor (2008), but increased use the requirement management activities is necessary.

CMMI level 2 has established a basic project management processes, which are schedule and functionality, track, and cost. The CMMI level 2 activities are broken down into three main activities to achieve the requirement management. According to Zainol and Mansoor (2008), the Malaysian companies always or regularly conduct to the first activity. Also, the outcome percentage for interested or use the first activity of CMMI level 2 in UUM IT is high. Hence, these sub-activities are performed among the Malaysian companies, even in UUM IT.

Furthermore, the result of second activity of CMMI level 2 shows that most Malaysian companies always or regularly conduct these sub-activities; which is similar to the outcome for CMMI level 2 of UUM IT. Besides, Zainol and Mansoor

(2008) state that the outcome of the third activity of CMMI level 2 shows that majority of the Malaysian companies were performing both these sub-activities regularly or always; which similar or near to the percentage of the current study. Even though, that the percentage of use the activities of CMMI level 2 are high, but the percentage of heard or used CMMI level 2 is low. This fact refers that is a need to awareness the developers about use CMMI level 2.

In the above discussion, comparison results indicated that the RM in previous study is not having a proper approach for managing in the companies of Malaysia software industry. Unfortunately, this fact has been confirmed through this study about the UUM IT. Furthermore, Malaysian software engineers have lack to use best requirements management practices, this fact is confirmed in UUM IT, where only 6 from developer team have heard of CMMI. Thus, UUM IT should be introduced a definite requirements management approach to UUM IT developer team.

4.9 Summary

This study has given us good indication of how the UUM IT manages their software projects. However, this study has limitation that should be highlighted here. Actually, the result of this study is analyzed based on only 23 developer of UUM IT. Furthermore, the study has revealed the current state of RM in the UUM IT. Moreover, in order to support the UUM IT to employ best practices of RM, a well defined RM process should be introduced to them. By having the workflow

integration, the tools will provide a complete and comprehensive guide to the developers on how to manage their RM and provide best practices of RM.

CHAPTER FIVE
DISCUSSION AND CONCLUSION

This chapter discusses results of the study depending on the outcome of the survey which conducted in the UUM IT for assessment carried out on CMMI level 2. The chapter is divided into three sections. The first section discusses the achievement of the objectives for the current study. The second section discusses the vision for the future work of the current study. The final section details summarize of the whole study.

5.1 Discussion

Good requirements engineering practices can either reduce the cost of the development project or increases the quality of the project (Solemon et al., 2010). The current study aims to investigate the current situation for the requirements management practices in UUM IT. The outcome refers that the describing of the requirements through use standard templates/documents has been used widely in UUM IT, where the percentage has been increased to reach 65%. On other hand, Zainol and Mansoor (2008) emphasizes that the UML is not regularly used to describe requirements in the companies, this fact confirm by our study, while the UUM IT developer team sometimes use UML to describe requirements. Moreover, the companies are regularly described requirements with natural language, which are similar with the current study. The companies are sometimes specifying their requirements with unique identification; this situation applies to UUM IT also. In

general, the outcome of the current study shows that there is a similar with the previous study, which is a weakness point in requirements management development, but there is little improvement.

Zhang, Li, and Liu (2014) refer that the requirement management activities can improve the efficient and effective of design work for software projects. This study used mixed method approach for more reliability outcome, where analyzing the result depend on questionnaire with UUM IT team, and interview with managers of UUM IT team. The outcome of study confirmed that the UUM IT has employed all requirement management activities with percentage around 50% in UUM IT. This fact has confirmed by the managers. Thus, it is an indication that there is interest in the use of requirements management activities in UUM IT. In details, before starting new projects, the UUM IT always carries out feasibility studies. On other hand, manage requirements through using database in the Malaysian companies are rarely, as well as, reuse requirements over different projects. This fact has confirmed in the UUM IT, where 34% of the UUM IT developer team never use database to manage their requirements. Although, 19% Malaysian companies tend not apply requirements management or limited, but Zainol and Mansoor (2008) argues it is still positive with four-fifths of the Malaysian companies were understand the significance of requirement management activities. This percentage has increased to become 21% in th UUM IT, even though we agree about the positive point

for Zainol and Mansoor (2008), but increased use the requirement management activities is necessary.

There are different models can use helps to improve the requirements management activities, such as CMMI. CMMI level 2 is a common model proved its worth. This study evaluate the relationship among the CMMI level 2with requirements managements activities as positive relationship; where increase used CMMI level 2 helps to improve the requirements managements activities. The CMM level 2 activities are broken down into three main activities to achieve the requirement management. According to Zainol and Mansoor (2008) the Malaysian companies always or regularly conduct to the first activity. Also, the outcome percentage for interested or use the first activity of CMMI level 2 in UUM IT is high. Hence, these sub-activities are performing from the Malaysian companies, even UUM IT.

Furthermore, the result of second activity of CMM level 2 shows that most Malaysian companies always or regularly conduct these sub-activities; which is similar to the outcome for CMMI level 2 of UUM IT. Beside, Zainol and Mansoor (2008) refers that the outcome of the third activity of CMM level 2 shows that majority of the Malaysian companies were performing both these sub-activities regularly or always; which similar or near to the percentage of the current study. Even though, that the percentage of use the activities of CMMI level 2 are high, but the percentage of heard or used CMMI level 2 is low. This fact refers that is a need to awareness the developers about use CMMI level 2.

5.2 Recommendations

The final conclusion of the study on requirements management activities, carried out by UUM IT, was that there were two main recommendations should be mentioned:

1. The UUM IT team needs to awareness and truing about the importance of using requirements management activities through holding seminars and workshop, and so on.
2. The UUM IT team should more pay attention the use of successful requirements management activities model, such as CMMI.

5.3 Future Work

The intention of the CMMI level 2 is providing a systematic way to conduct RM practices. The current study attempted to investigate the use of requirements management activities within higher educational institutions through UUM IT. This study showed a vision for the future work as follows:

i. In this study, the case study presented to assess the use of CMMI level 2 in one university in Malaysia, which cannot be generalized. Therefore, similar study should also be conducted in other universities in Malaysia, in order to know the use of CMMI level 2 within their organization.

ii. The percentage of heard or used CMMI level 2 is low, which means there are a need to research deeply to understand the reasons reluctance the IT developers to use these methodologies.

5.4 Limitation of the Study

The limits of this study focused on:

1. The RM practice within the boundaries of RE, as well as, CMMI level 2 which is the management level.
2. The study focused on university computer services provider, specifically UUM IT.

5.5 Conclusion

The current study illustrated the situation of requirements management in the UUM IT. This study revealed that hard to identified appropriate approach of managing requirements to the UUM IT; because of lack of using best requirements management practices between the UUM IT developer team. The result of the study confirmed that CMMI model not really familiar with the UUM IT developer team. Thus, the current study gives a good motivation to understand the reasons to failure experiment of use requirements management in the future.

REFERENCES

Abran, A., & Moore, J. W. (2001). *Guide to the Software Engineering Body of Knowledge, Trial Version*. Los Alamitos,CA: IEEE Computer Society Press.

Ahonen, J. J., & Savolainen, P. (2010). Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects. *The Journal of Systems and Software*, 2175–2187.

Anderson, S., & Felici, M. (2001). Requirements engineering questionnaire. *vol*, *1*, 15.

Asghar, S., & Umar, M. (2010). Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components Sohail.Asghar@jinnah.edu.pk Mahrukhumar@yahoo.com important tasks . Entire software is supported by four pillars of requ. *International Journal of Software Engineering (IJSE)*, *1*(2), 32–50.

Basili, V. R., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., Rombach, D., et al. (2013).Linking software development and business strategy through measurement. *IEEE Computer, 43*(4), 57-65.

Blanchard, B. S., & Fabrycky, W. J. (2010). Systems engineering and analysis, Prentice Hall.

Calinescu, R., Ghezzi, C., Kwiatkowska, M., &Mirandola, R. (2012). Self-adaptive software needs quantitative verification at runtime. *Communications of the ACM*, *55*(9), 69-77.

Carstens, D. S., Richardson, G. L., & Smith, R. B. (2013). *Project Management Tools and Techniques: A Practical Guide*. CRC Press.

Casallas Gutiérrez, R., & Arboleda Jiménez, H. F. (2011). QualDev process: procesos adaptables de desarrollo de software para proyectos ágiles. *Revista Ingeniería y Competitividad*, *6*(2), 93-104.

Cerpa, N., & Verner, J. M. (2009). Why Did Your Project Fail ? *Communications of the ACM*, *52*(12), 130–134.

Charette, B. R. N. (2005). We waste billions of dollars each year on entirely preventable mistakes. *IEEE Spectrum*, (September 2005), 42–49.

Chrissis, M. B., & Weber, C. V. (1993). Capability Maturity Model for Software, Version 1.1. *Software Engineering Institute-Carnegie Mellon University. Pittsburgh, Pennsylvania*.

Chua, Y. P. (2013). *Mastering research statistics*. Mcgraw-Hill Education.

Crespo, D., & Ruiz, M. (2012).*Decision making support in CMMI process areas using multiparadigm simulation modeling.* Paper presented at the Proceedings of the 2012 Winter Simulation Conference (WSC), Berlin,(pp. 1-12). IEEE.

Cuevas, G., & Serrano, A. (2004). Assessment of the Requirement Management Process using a Two-Stage Questionnaire. In *Proceedings Of the Fourth International Conference on Quality Software* (pp. 110–116). Los Alamitos,CA: IEEE Computer Society

Dalcher, D. (2009). Software Project Success: Moving Beyond Failure. *Experiences and Advances in Software Quality*, *5*.

Damian, D., Zowghi, D., Vaidyanathasamy, L., Pal, Y., Drive, H. B., & Samy, L. (2007). An Industrial Experience in Process Improvement : An early assessment at the Australian Center for Unisys Software Faculty of Information Technology University of Technology, Broadway Unisys Australia Limited 3.

Davis, A. M., & Leffingwell, D. A. (1996). Using Requirements Management to Speed Delivery of Higher Quality Applications. Rational White Paper.

Dutoit, A., & Paech, B. (2000). Rationale Management in Software Engineering. In : Handbook of Software Engineering and Knowledge.

Emam, K. E., & Birk, A. (2000). Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability. *IEEE Transactions on Software Engineering*, *26*(6).

Firesmith, D. G. (2007). Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them. *Journal of Object Technology*, *6*(1), 17–33.

Gallagher, V. C., Gallagher, K. P., & Kaiser, K. M. (2013). Mid-Level Information Technology Professionals: Skills and Traits Relevant to Fit, Individual and Organizational Success Factors. *International Journal of Social and Organizational Dynamics in IT (IJSODIT)*, *3*(2), 22-40.

Ge, N., & Pantel, M. (2012). Time properties verification framework for UML-MARTE safety critical real-time systems. In *Modelling Foundations and Applications* (pp. 352-367). Springer Berlin Heidelberg.

Gotel, O. C., Marchese, F. T., & Morris, S. J. (2008). The potential for synergy between information visualization and software engineering visualization. *12th International Conference in Information Visualisation, 2008. IV'08.* (pp. 547-552). IEEE.

Grenn, M. W. (2013). *A Theory of Information Quality and a Framework for its Implementation in the Requirements Engineering Process* (Doctoral dissertation, The George Washington University).

Hansen, S., Berente, N., & Lyytinen, K. (2009). Requirements in the 21st Century : Current Practice and Emerging Trends *. In *Design Requirements Engineering: A Ten-Year Perspective* (pp. 44–87). Springer Berlin Heidelberg.

Heidenheimer, A. J., Heclo, H., & Adams, C. T. (1983). Comparative Public Policy. New York: St. *Martin's*, 248-51.

Hull, E., Jackson, K., & Dick, J. (2011). *Requirements engineering*. Springer Science & Business Media.

Jarke, M. (1998). Requirements Tracing. *Communications of the ACM*, *41*(12), 32–36.

Jones, C. (1996). Strategies for Managing Requirements Creep. *IEEE Computer*, *29*(6), 92–94.

Joseph, D., Ang, S., Chang, R. H., & Slaughter, S. A. (2010). Practical intelligence in IT: assessing soft skills of IT professionals. *Communications of the ACM*, *53*(2), 149-154.

KarimJallow, A., Demian, P., N. Baldwin, A., &Anumba, C. (2014).An empirical study of the complexity of requirements management in construction projects.*Engineering, Construction and Architectural Management*, *21*(5), 505-531.

Khan, M. N. A., Khalid, M., &ulHaq, S. (2013). Review of requirements management issues in software development. *International Journal of Modern Education and Computer Science (IJMECS)*, *5*(1), 21.

Khankaew, S., & Riddle, S. (2014). A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand. In *Empirical*

Requirements Engineering (EmpiRE), 2014 IEEE Fourth International Workshop on (pp. 1-8). IEEE.

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT Press.

Kragelund, J. (2012). Advanced tool support for requirements engineering. *Master's thesis, Technical University of Denmark*.

Kulak, D., & Guiney, E. (2012). *Use cases : requirements in context*. Addison-Wesley.

Kulpa, M. K., & Johnson, K. A. (2004). *Interpreting the CMMI (R): A Process Improvement Approach*. CRC Press.

Lee, K. H., Min, P. G., Cho, J. H., & Lim, D. J. (2012). Model-driven requirements validation for automotive embedded software using UML. In *Computing Technology and Information Management (ICCM), 2012 8th International Conference on* (Vol. 1, pp. 46-50). IEEE.

Leffingwell, D., & Widrig, D. (2000). *Managing Software Requirements: A Unified Approach*. Boston MA: Addison-Wesley.

Li, W., Brown, D., Hayes, J. H., & Truszczynski, M. (2014). Answer-Set Programming in Requirements Engineering. In *Requirements Engineering: Foundation for Software Quality* (pp. 168-183). Springer International Publishing.

McLeod, L., & McDonell, S. (2011). Factors that affect software systems development project: a survey of research. *ACM Computing Surveys*, *43*(4).

Mizouni, R., & Lazarova-Molnar, S. (2012). Simulation-based feature selection for software requirements baseline. *Journal of Software*, *7*(7), 1440-1450.

Nasir, M. H. N., & Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. *Scientific research and essays*, *6*(10), 2174-2186.

Nikula, U., Sajaneimi, J., & Kalviainen, H. (2000). *Survey on RE in Small-and-Medium Enterprises*.

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research process. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (p. 7).ACM.

O'Regan, G. (2010). *Introduction to software process improvement*. Springer Science & Business Media.

Othman, M., Zain, A. M., & Hamdan, A. R. (2010). A Review on Project Management and Issues Surrounding Dynamic Development Environment of ICT project: Formation of Research Area. *JDCTA*, *4*(1), 96-105.

Park, S., & Nang, J. (1998). Requirements management in large software system development. *IEEE International Conference on Systems, Man, and Cybernetics*, (Vol. 3, pp. 2680-2685). IEEE.

Park, S., & Nang, J. (2004). Requirements Management in Large Software System Development. In *Proceedings of IEEE International on System, Man and Cybernatics*. IEEE Computer Society.

Pohl, K. (2010). *Requirements engineering : fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated.

Pohl, K. (2013). The three dimensions of requirements engineering. In *Seminal Contributions to Information Systems Engineering* (pp. 63-80). Springer Berlin Heidelberg.

Pohl, K. (2013). The three dimensions of requirements engineering. In *Seminal Contributions to Information Systems Engineering* (pp. 63-80). Springer Berlin Heidelberg.

Product Development Team. (2002a). *Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation* (p. CMU/SEI–2002–TR–012). Pittsburgh.

Product Development Team. (2002b). *Capability Maturity Model Integration (CMMI),Version 1.1,Continuous Representation* (p. CMU/SEI–2002–TR–011). Pittsburgh.

Purbasari, A., Iping, S. S., Santoso, O. S., & Mandala, R. (2013, July). Designing Artificial Immune System Based on Clonal Selection: Using Agent-Based Modeling Approach. In *Modelling Symposium (AMS), 2013 7th Asia* (pp. 11-15). IEEE.

Reddivari, S. (2013). Visual analytics for software requirements engineering. In *Requirements Engineering Conference (RE), 2013 21st IEEE International* (pp. 389-392). IEEE.

Regnell, B., Svensson, R. B., & Wnuk, K. (2008). Can we beat the complexity of very large-scale requirements engineering?. In *Requirements Engineering: Foundation for Software Quality* (pp. 123-128). Springer Berlin Heidelberg.

Robertson, S., & James, R. (2012). *Mastering the requirements process: getting requirements right*. Addison-Wesley.

Rose, R. (2004). *Learning from comparative public policy: A practical guide*. Routledge.

Rosenkranz, C. (2012). The Emergence of Shared Understanding: Applying Functional Pragmatics to Study the Requirements Development Process. *Information Systems Journal*.

Saleem, H., Khan, M. Z. A., & Afzal, S. (2012). Towards Identification and Recognition of Trace Associations in Software Requirements Traceability. *perspectives*, *2*, 6.

Schulze, S., & Pretorius, L. (2013). An Exploratory Case Study on the Requirements Business Processes of a Typical South African High Technology Systems Engineering Company. *SAIEE Africa Research Journal*, South African Institute of Electrical Engineers.*104*(1), 11-21.

Singer, L., Sim, S., & Lethbridge, T. (2008). Software Engineering Data Collection for Field Studies. In *Guide to Advanced Empirical Software Engineering* (pp. 9–34). London: Springer Publishing Company, Incorporated.

Sommervile, I. (2010). *Software Engineering* (9th Editio.). Addison-Wesley

Tahir, A., & Ahmad, R. (2010). Requirement Engineering Practices - An Empirical Study. *2010 International Conference on Computational Intelligence and Software Engineering*, 1–5. doi:10.1109/CISE.2010.5676827

Team, C, P. (2006). *CMMI for Development, version 1.2*.

Teijlingen, E. R. van, & Hundley, V. (2001). The importance of pilot studies. *Social Research Update*, (35).

Thayer, R. H., & Dorfman, M. (1990). *System and Software Requirements Engineering*. Los Alamitos,CA: IEEE Computer Society Press.

Wahono, R. S. (2003). I j w - 2 0 0 3. In *Proceedings of the IECI Japan Workshop 2003* (Vol. 5, pp. 55–58).

Weber, C. V., Curtis, B., & Chrissis, M. (1995). *The capability maturity model: Guidelines for improving the software process* (Vol. 441). Reading, MA: Addison-wesley.

Wiegers, K. E. (2003). *Software Requirements* (Second Edi.). Redmond Washington: Microsoft Press.

Wieringa, R. J. (2014). Requirements Specification. In *Design Science Methodology for Information Systems and Software Engineering* (pp. 51-57). Springer Berlin Heidelberg.

Yadav, M., & Kumar, K. (2014). CMM or CMMI, which is more appropriate for a software industry?. *International Journal in IT & Engineering*, *2*(1), 24-33.

Young, R. . (2004). *The Requirements Engineering*. Boston: Artech House.

Yu, A. T. W., & Geoffrey, S. Q. . (2013). Problems and solutions of requirements management for construction projects under the traditional procurement systems. *Facilities*, *31*(5/6), 223–237.

Zainol, A., & Mansoor, S. (2008). Investigation into Requirements Management Practices in the Malaysian Software Industry. In *Proc of International Conference on Computer Science and Software Engineering* (pp. 292–295). Wuhan, China: IEEE Computer Society.

Zhang, Z., Li, X., & Liu, Z. (2014). A closed-loop based framework for design requirement management. In Moving Integrated Product Development to Service Clouds in the Global Economy: Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering, September 8–11, 2014 (Vol. 1, p. 444). IOS Press.