*International Journal of Software Engineering & Research Methodology – IJSERM*
*Volume 1 : Issue 1*
*Publication Date : 09 January 2014*

# A Requirements Model for Requirements Management Tool to Support Small and Medium Projects

Azida Zainol
School of Computing, College of Arts and Sciences,
Universiti Utara Malaysia
06010 Sintok, Kedah, Malaysia
azida@uum.edu.my

Sa'ad Mansoor
School of Computer Science,
Bangor University,
Dean Street, LL57 1UT Gwynedd, United Kingdom
s.mansoor@bangor.ac.uk

*Abstract*—Presently, there are many requirements management tools available in the market. However, these tools are expensive, complicated, and difficult to learn as well as too sophisticated for small and medium projects. Thus, this paper attempts to describe a model of a requirements management tool that is suitable for small and medium projects (RMT-SMP) as well as suits for the Malaysia software industry. The model is known as requirements model which provides a complete guidance to designer and engineers. The research method involves the process of identifying and gathering the elements for the requirements management tool. After that, the elements are analyzed in the requirements analysis and modeling phase. In this phase, the requirements are identified and documented using UML and requirements specification. The requirements model which consists of several UML diagrams are developed to visualize the requirements for the tool. The requirements model is verified by using a prototype. The prototype is introduced to the respondents who are the software engineers and they manage to evaluate the prototype using perceived usefulness and ease of use (PUEU) questionnaire. As a result, the deliverables from this work are a comprehensive requirements model and supported by requirements specifications. Generally, this paper contributes to analysis phase of software development life cycle and specifically in the area of requirements model. The requirements model presented in this paper is a baseline for the development of a requirements management tool to support small and medium projects in the Malaysian software industry.

Keywords— Software requirements, software engineering, requirements model, requirements management, requirements management tool.

## I. INTRODUCTION

The sales of requirements management tools have been growing steadily in recent years [1]. There are many requirements management tools available in the market. It ranges from complicated and sophisticated tools to easy tools, from expensive tool to even a cheap or free tool. There are many requirements management tools in the market that claim to support the requirements management activities [2]. However, not all of these tools in the market are focus solely on requirements management activities. This is supported by a finding of a survey of requirements engineering tools by [3] that concluded the RE tools did not completely support requirements management activity.

The use of requirements management tools has become essential by considering the size and complexity of development efforts [4]. However, a study [5] of a survey in the Malaysian software industry revealed that there is no appropriate approach of managing requirements in small and medium projects. In addition, it is also reported that there is lack of using best requirements management practices among software practitioners.

There are some commercial off-the shelf-requirements management tools such as DOORS and Rational Requisite Pro. However, these tools use different concepts; have different capabilities and differing degrees of maturity with respect to their applicability in system engineering projects [6]. In addition, there is an open requirements management tool developed by Etish software development team on Eclipse platform, known as Useme [7]. This tool is a collaborative tool that facilitates the utilization of industry standards and best practices in requirements capture. Nevertheless, this tool focuses on requirements capture, while the need in the Malaysian software industry is a tool that is able to manage requirements management activity.

A study by [8], only 12.2% out of 74 respondents used requirements management tools, which is 10.8%, claimed used Rational Requisite Pro and 1.4% used other type of requirements management tools. Almost all respondents never used the requirements management tools to support their software development project, even in small and medium projects. It is also reported by [9] that the industry is lack of using sophisticated tools. If no significant improvement and progress to overcome these problems, this phenomenon will be one of the major challenges in software engineering in the Malaysian software industry.

One of the major tasks in order to overcome those problems is to introduce an open source requirements management tool purposely for the Malaysian software industry. The tool is known as Requirement Management Tool

for Small Medium Projects (RMT-SMP) which automate the best practices of requirements management activities in order to have a better approach and practices of software engineering. Hence, this paper aims to present the requirements model of RMT-SMP using Unified Modeling Language (UML).

Requirement modeling is an established method for detecting defects in requirement specifications [10]. The models are used to discover and clarify the functional and data requirements for software system. It is the basis for understanding user requirements. This paper contributes to the analysis phase of software development life cycle in general as well as in specific area of requirements model.

The work presented in this paper begins with section 2, which describes about reviews. Section 3 elaborates the research methods. Section 4 presents the result and section 5 concludes this work.

## II. REVIEWS

### *A. Requirements Model*

Software engineering activity begins with a series of modeling tasks that lead to a specification of requirements and a design representation for the software to be built. According to [11], the requirement model consists of a set of models is the first technical representation of a system. These models visualize the system from different aspects of a system. The requirements model is also a reference to assist in the development of a system during various stages of its life cycle. Requirements modeling are characterize as an information technology to capture, communicate, track, analyze, verify, validate, view, manage hierarchical and interrelated requirements for a system [12].

In modeling the requirements, the primary focus is on what rather than how. This is because the requirements model should achieve the following objectives [11]:

- to describe what the customer requires
- to establish a basis for the creation of a software design
- to define a set of requirements that can be validated once the software is built

### *B. Requirements Management*

The good practices in requirements engineering is to ensure all the requirements are documented and manage properly. Thus, requirements management is an important activity that should not be ignored. According to [13], requirements management is a set of activities that help the project team identify, control, and track requirements and changes to requirements at any time as projects proceeds. While [14] stated that requirements management focus on maintaining traceability, the ability to describe and follows the life of a requirements. Prior research also highlighted the importance of having a proper requirements management. As stated by [11] and [14], it can be concluded that requirements management is an ongoing activity that should be perform throughout the software life cycle.

As the project progress, [15] mentioned that requirements management includes activities to maintain the integrity, accuracy and currency of the requirements agreement. Thus, the requirements management involve the following activities [15]:

- control changes to the requirements baseline
- keeping project plans current with the requirements
- controlling versions of both individual requirements and requirements documents
- tracking the status of the requirements in the baseline
- managing the logical links between individual requirements and other project work products

The process of managing requirements becomes easy and more practical when using a software tool. A requirements management tool that stores information in a multiuser database provides a robust solution. In addition, [15] clearly clarifies the following benefits of using requirements management tool;

- manage versions and change
- store requirements attributes
- facilitate impact analysis
- track requirements status
- control access
- communicate with stakeholders
- reuse requirements

The sales of requirements management tools have been growing steadily in recent years [16]. There are many requirements management tools available in the market. It ranges from complicated and sophisticated tools to easy tools, from expensive tool to even a cheap or free tool. There are many requirements management tools in the market that claim to support the requirements management activities [2]. However, not all of these tools in the market are focus solely on requirements management activities. The tools available in the market are developed by the vendors in order to manage all the requirements generally. Nevertheless, each company has their own culture and policy towards managing requirements. Thus, this paper attempts to capture the requirements for a requirements management tool that support small projects in the Malaysian software market. The requirements captured will be presented using requirements model in order to perceive inspirations.

## III. RESEARCH METHOD

The approach for requirements modeling is adopted from [11] that describe four elements of requirements model. These elements are scenario-based models, class models, behavioural models and flow models. These models present the problems from a different point of view. Furthermore, the model proposed in this paper is using Unified Modeling Language (UML) [17] as modeling technique to visualize the requirements. UML is a standard language for writing software blueprints. It may be used to visualize, specify, construct and document the artifacts of a software-intensive system [18].

In identifying the elements of the requirements management tool, an investigation of software engineering and requirements management practices were conducted in the Malaysian software market using questionnaire. This is important to understand the market as well as gathering the elements of a requirements management tool should have in order to fit the market demand. On the other hand, a comprehensive theoretical study is also conducted in order to complete the elements that the tool should have. After that, the elements are analyzed and being compared with 15 best available requirements management tools in the market.

The next phase is requirements analysis and modeling. In this phase, the elements, as an input are being analyzed thoroughly in order to identify the requirements of a requirements management tool. The requirements are documented as specifications as well as visualize it into requirements model. The requirements model is composed of a set of UML diagrams that document the requirements from different level of abstractions. In order to ensure that the requirements model is accurate, it is important to verify it using a prototype. Thus, a prototype that reflects the requirements model is developed in Eclipse [19]. The model was evaluated to ensure that the user requirements are met. The verification phase is achieved by allowing few companies using the prototype during the software development phase. A questionnaire was developed from the perceived usefulness and ease of use (PUEU) questions [20]. PUEU instrument focuses on two important components namely perceived usefulness and perceived ease of use. It involved twelve questions with scale from 1 to 5. Scale 1 equivalent to strongly disagree, scale 2 equivalents to agree, scale 3 equivalents to neutral, scale 4 equivalents to disagree and scale 5 equivalents to strongly agree. Moreover, [20] concluded that perceived usefulness is a strong correlate of user acceptance and should not be ignored by those attempting to design or implement successful systems. Altogether, there are 10 respondents involved in evaluating the prototype. The evaluation begins with a demonstration of how to use the tool. The prototype was explained to the respondents thoroughly in terms of functionality and capability. After they used it, the questionnaire was distributed for them to evaluate the prototype.

## IV. RESULT

This section discusses the result of the study that can be a guideline in developing a requirements management tool.

### *A. Use case diagram*

A use case diagram is used to show what system functions are performed for which actor [21]. It describes the functionality of a system from the perspective of users in the system [22]. Fig. 1 below is a use case diagram that visualizes the functional requirements of a requirement management tool. It can be seen that there are 10 use cases that also supplemented with 10 use case specifications. The use cases are : Manage requirements identification, manage requirements classifying and viewing, manage requirements baselining, manage change control, manage version control, manage status tracking, manage requirements tracing, manage document importing, manage document generation and manage tailoring and extensibility.

### *B. Use case specifications*

The use case specification is supplementary documents that describe each use case in detail using natural language. The specification could be developed using specific standard in order to ensure each scenario is described completely. From the use case diagram, there is 10 use cases, meaning that there are 10 use case specifications been described comprehensively.

### *C. Activity diagrams*

In UML, an activity diagram is used to display the sequence of activities. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modeling where they are used for detailing the processes involved in business activities. In this thesis, the activity diagram is used to visualize the sequence of activity for a use case. Thus, there are 10 activity diagrams were developed for this study.

### *D. Class diagram*

The class diagram shows the building blocks of any object-orientated system. Class diagrams depict a static view of the model, or part of the model, describing what attributes and behavior it has rather than detailing the methods for achieving operations. Class diagrams are most useful in illustrating relationships between classes and interfaces as shown in Fig.2 below.

### *E. Package diagram*

The Package diagrams are used to categorize and group classes in a manner that makes them more manageable for larger system. The categorization can be applied to various elements of analysis model such as use cases and analysis classes. Thus, there a package diagram as shown in the Fig. 3 below.

### *F. Collaboration diagrams*

Collaboration diagram also know as communication diagram provides an indication of the temporal order of communications that emphasize on the relationships among the objects and classes. In this study, there are 30 number of collaboration diagrams that is part of requirements model

### *G. State diagrams*

State diagrams model an object's states, the actions that are performed depending on those states, and the transitions between the states of the object. The state diagram is used to model the state of each class as the system performs its function and the states of the system as observed from the outside as the system performs its function. In this study, there are 5 states diagrams to represent the requirements model.

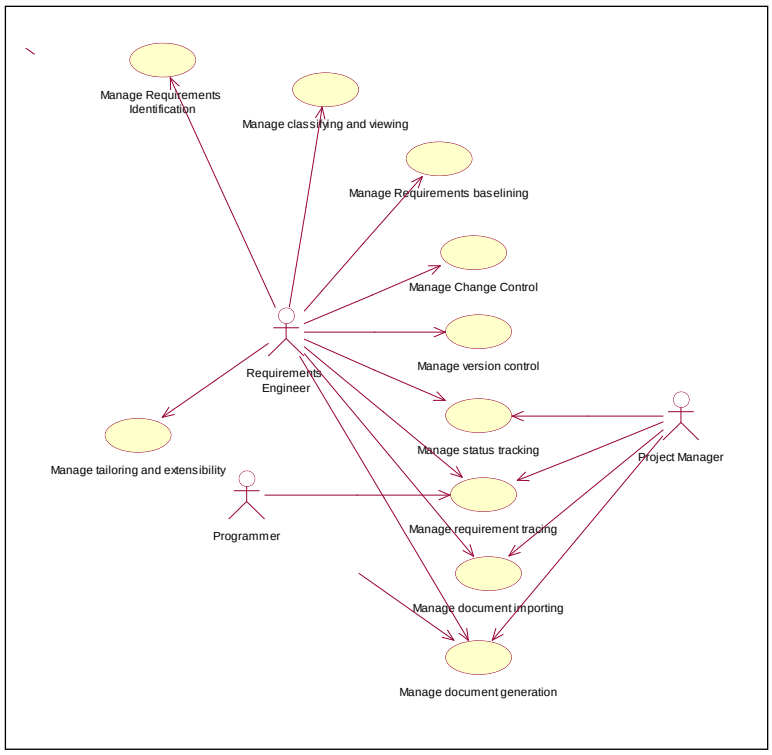


Figure 1. Use case diagram

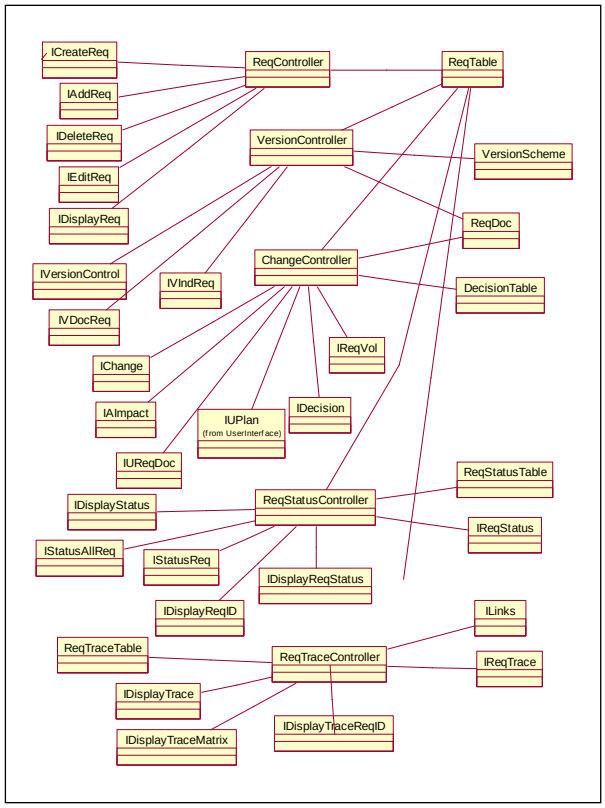


Figure 2. Class diagram

### *H. Sequence diagrams*

Sequence diagram depicts the dynamic communications between objects during execution of a task. It shows the temporal order in which messages are sent between the objects to accomplish that task. In addition, sequence diagram is also used to show the interactions in one use case or in one scenario of a software system. For this study, a sequence diagram is used to depict each scenario in every use case. Thus, there are 30 sequence diagrams for modeling a requirements management tool.

### *I. Prototype Verification*

The prototype is developed purposely based on the requirements model. The intention of this prototype verification is to test the behavioral construct of perceived usefulness and perceived ease of usage acceptance of requirements model tool. The prototype is very useful in overcoming misunderstanding or ambiguity requirements. It also allows the user to identify the features of the proposed tool that they need to be improved.

The result reveals high levels of perceived usefulness and perceived ease of use. It also shows positive trends in answering the questionnaire because most of the respondents choose between strongly agree and agree scale. While the percentage of respondents who answered strongly disagrees or disagrees is 0 percent.

Table I below shows the frequency of scored for perceived usefulness. It can be seen that 80% of the respondents are positively agree that the tool is useful to them. It is reported that 20% of the respondents are neutral about the productivity and effectiveness of their job whenever they used this tool.

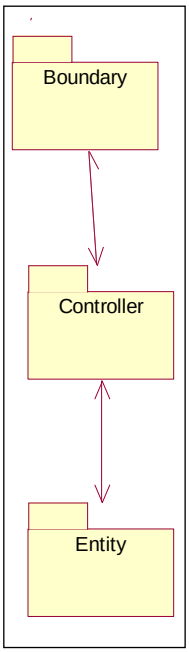


Figure 3. Package diagram

Table II depicts the frequency of scored of perceived ease of use. It shows that almost everybody considers the tool is easy to use.

TABLE I. PERCEIVED USEFULNESS FREQUENCY

| Questions/ Scale | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| Strongly Agree | 40% 4 | 30% 3 | 40% 4 | 30% 3 | 30% 3 | 40% 4 |
| Agree | 60% 6 | 0% 5 | 40% 4 | 50% 5 | 70% 7 | 60% 6 |
| Neutral | 0 | 20% 2 | 20% 2 | 20% 2 | 0 | 0 |
| Disagree | 0 | 0 | 0 | 0 | 0 | 0 |
| Strongly Disagree | 0 | 0 | 0 | 0 | 0 | 0 |

TABLE II. PERCEIVED OF EASE OF USE

| Questions/ Scale | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 |
|---|---|---|---|---|---|---|
| Strongly Agree | 60% 6 | 50% 5 | 40% 4 | 30% 3 | 30% 3 | 50% 5 |
| Agree | 40% 4 | 50% 5 | 60% 6 | 70% 7 | 60% 6 | 50% 5 |
| Neutral | 0 | 0 | 0 | 0 | 10% 1 | 0 |
| Disagree | 0 | 0 | 0 | 0 | 0 | 0 |
| Strongly Disagree | 0 | 0 | 0 | 0 | 0 | 0 |

## V. Conclusions

This study has shown a comprehensive requirements model for RMT-SMP that provides guideline in developing the tool in the future. The requirements model that is consists of several diagrams using UML will be transform into design and very helpful during implementation phase afterwards. Without this guideline, the development of a requirements management tool is prone to requirements deficiency which leads to development failure, wasting cost, time and resources.

The work presented in this paper represents the findings from limited respondents. Since there is only 10 respondents from 3 companies, the result here is not illustrates the user needs from the Malaysian software market. In addition, the requirements model proposed in this paper is only verified using prototype. It is not validated using any Computer Aided Software Engineering tools to check on model consistency, completeness and any other possible variables. Yet, the result of prototype verification that focuses on perceived usefulness and perceived of ease of use give us a motivation to further research on designing and implementing a requirements management tool.

In the future, this study can be expanded to many companies in the Malaysian software industry. Thus, we can capture more requirements to illustrate the software industry needs. However, by having this requirements model is an initial stage for us to design and develop a requirements management tool that perhaps suitable for assisting the Malaysian software companies during their software development.

## *References*

[1] The Standish Group International. :Special COMPASS Report on Requirements Management Tool. (1998)

[2] Lang, M., Dunggan, J.: A Tool to Support Collaborative Software Requirements Management. Requirements Engineering, 6(3), 161-172. (2001)

[3] Carrillo de Gea, J.M.; Nicolas, J.; Aleman, J.L.F.; Toval, A.; Ebert, C.; Vizcaino, A. : Requirements Engineering Tools. IEEE Journal Software, 28 (4), 86 – 91, 2011

[4] Hammer, T., & Huffman, L.: Automated Requirements Management – Beware HOW You Use Tools: An Experience Report. In: Proceedings of the Third International Conference on Requirements Engineering (pp. 43-40). Los Alamitos,CA (1998)

[5] Zainol, A., Mansoor, S. : Investigation into Requirements Management Practices in the Malaysian Software Industry. In: Proc of International Conference on Computer Science and Software Engineering, pp. 292--295. IEEE Computer Society (2008)

[6] Hoffmann, M., Kuhn, N., Weber, M., Bittner, M.: Requirements for Requirements Management Tools. In: Preceedings of 12th IEEE International Requirements Engineering Conference, pp. 301-308, (2004)

[7] Etish project, http://www.etish.org/useme/about.html

[8] Zainol, A., Mansoor, S.: A Survey of Software Engineering Practice in the Software Industry. In: Proc of IASTED International Conference on Software Engineering (2009)

[9] Zainol, A., Mansoor, S.: An Investigation of a Requirements Management Tool Elements In: Proc of IEEE Conference on Open System 2011. IEEE Computer Society (2011)

[10] Wohlin, C., Runeson, P., Ohlsson, M. C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers (2000)

[11] R.S.Pressman, Software engineering : a practitioner's approach, seventh edition (Singapore : McGraw-Hill, 2009)

[12] D. Baker, Requirements Modeling Technology : A Vision for Better and Cheaper System, *IEEE Database*, 2007, 3-6.

[13] Maciaszek, L. A. (2007). *Requirements Analysis and Design*. Harlow: Addison-Wesley.

[14] A. Finkelstein & W.Emmerich, The Future of Requirements Management Tools,*Information Systems in Public Management and Law*, 2000

[15] K. E. Wiegers, *Software requirements* (Redmond Washington : Microsoft Press, 2003)

[16] Standish Group International, Special COMPASS report on requirements management tool. West Yarmouth, MA, USA, 1998.

[17] Object Management Group-UML, www.uml.org

[18] G. Booch, I. Jacobson & J. Rumbaugh, *The Unified Modeling Langguage User Guide*, (Boston : Addison-Wesley, 2001)

[19] Eclipse Community, www.eclipse.org

[20] F.D. Davis, Perceived usefulness, Perceived ease of use, and User Acceptance of Information Technology, *MIS Quarterly*, 1989, 319-340.

[21] Dedeke, A., & Lieberman, B. (2006). Qualifying Use-Case Diagram Associations. *IEEE Computer*, 23-29

[22] Sengupta, S., & Bhattacharya, S. (2006). Formalization of UML Use Case Diagram-a Z Notation Based Approach. *Proceedings of International Conference on Computing & Informatics* (pp. 1-6). IEEE Computer Society.