

# An Investigation of a Requirements Management Tool elements

Azida Zainol

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

Sa'ad Mansoor

School of Computer Science  
Bangor University  
Gwynedd, United Kingdom  
s.mansoor@bangor.ac.uk

**Abstract**—Developing quality software is becoming challenges that required the practitioners to employ appropriate techniques, methodologies and good practices in software engineering. In the Malaysian software industry, the challenges are the practitioners are lack of practicing software engineering especially in the requirements management practices as well as the requirements management tool is not widely facilitated during software project development. In order to encourage them to overcome these problems, this paper attempts to recommend a requirements management tool that embeds the best practices in managing requirements. In addition, this paper aims to investigate the elements of the tool to guarantee that the tool is appropriate for the Malaysian software needs. Then, a comparative study is carried out to search for various available requirements management tools that have those elements. The result shown that, there is no specific tool that has all the elements in order to fit the Malaysian software needs. Therefore, it is essential to develop a requirements management tool that able to assist the software practitioners in developing quality software.

**Keywords**—component; requirements management; requirements management tool; software engineering; requirements engineering

## I. INTRODUCTION

Over the last decade, software engineering had a remarkable impact on society and the global economy. As in the United States, about USD275 billions are spent each year on approximately 200,000 application projects with a significant software component [1]. The software industry has driven about 1/6 of the GDP revenue in the United States. Products that have a major software component can be seen everywhere in our daily lives such as in computers, email, world-wide-web, cell-phones, washing machines and microwaves. It is predicted that this impact will continue to increase in the next decades [2].

Nevertheless, most of the problems that the software industry faces today are the same as the ones experienced in 1968, when the term “Software engineering” was first coined at the NATO conference in Garmisch, Germany [3]. A large number of software projects have failed or have been cancelled due to cost overruns, schedule delays and unsatisfied customers and stakeholders. Data from the Chaos Report of the Standish Group [4] shows that on average more than 23%

of projects fail and more than 49% of projects are challenged, i.e. late and/or over-budget. This phenomenon has thrown the whole software industry for almost four decades. Poor Requirements Engineering (RE) practice has been blamed as one of the major reasons contributing to this situation [5, 6].

As one of the processes in software engineering, RE plays a vital role to ensure the overall success of software projects. Requirements management (RM) is a part of RE activity that concentrating on handling the change management, traceability, version control and tracking the requirements status. These days, the requirements management activity is not entirely taken into consideration during software development. Having practicing requirements management during a software project development is the first step towards increasing the overall quality of a software product. Requirements are attributes that define the capability, characteristics, performance and quality characteristics of a system. In order to ensure the quality of software requirements specification, there needs to be a strong emphasis on implementing engineering disciplines into the RE process, including the requirements management activity by using various good practices, techniques and methodologies [7, 8, 9, 10].

Requirements have a tendency to change during system development and these changes must be managed. Usually, during RE process involves a large amount of data and unstable requirements. Thus, RM tools have been developed to help in managing those [11]. RM tools support the management of requirements database and changes to these requirements. They collect together the system requirements in a database or repository and provide a range of facilities to access the information about the requirements.

Nowadays, there are various requirements management tools in the market. It ranges from the light, easy and cheap tool to the complex, sophisticated and expensive tools. However, the question that arises is, do the available tools in the market suitable and appropriate for every software industry? In the Malaysian software industry, the challenges are the practitioners are lack of practicing software engineering especially in the requirements management practices [12]. Also, the requirements management tool is not widely conducted during software project development [13]. 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 find the feasible solution that would encourage the Malaysian practitioner to conduct and practice requirements management activity during software project development. Hence, this paper attempts to recommend a requirements management tool that embeds the best practices for requirements management activity in order to have a better approach and practices of software engineering. Moreover, this requirements management tool should be appropriate for the Malaysian software industry. Thus, this paper aims to investigate the characteristics or elements that a requirements management tool should have in order to become suitable for the Malaysian software industry. Then, these elements are compared with several requirements management tool characteristics. The investigation begins with section II that presents the reviews of requirements management tool. Then section III defines the requirements management tool elements from general and specific elements perspective. Section IV shows the comparative study of generic and specific elements. Finally, section V concludes the investigation.

## II. REVIEWS

Project teams traditionally document their requirements in a structured software requirements specification (SRS) written in natural language [14, 15]. However, the document-based SRS has following problems [14]:

- It is difficult to keep the documents current and synchronized.
- It is also difficult to communicate if there is any change because the communicating among the team members is a manual process.
- It is not easy to store supplementary information especially attributes about each requirements.
- It is also hard to define links between functional requirements and other system elements.
- Tracking requirements status is very hard and involve manual process
- It is also difficult for team members to modify the requirements, if the team members are geographically separated.
- There is also no convenient place to store proposed requirements that were rejected and requirements that were deleted from a baseline.

It can be concluded that there are many limitations in document-based SRS. Thus, a requirements management tool that stores information in a multi-user database provides a robust solution towards these limitations. Using a database could benefit the large projects as well as the small projects.

For a small project, the central repository could be the spreadsheet applications or simple databases to manage the requirements, storing relevant information, text as well as attributes of each requirement. Relational databases are used for storing and managing large number of records, which have the same structure and minimal links between them. Currently

many RM tools are based on relational databases. According to [9], it is possible to maintain many links with a relational database, but it is insufficient as it requires operations on several different tables. Thus, in [9] stated that object-oriented databases are structurally more suited to requirements management. They allow different types of information to be maintained in different objects and the way they manage links between objects is fairly straightforward.

Moreover, the larger project could employ the commercial requirements management tools available in the current market. The commercial tools provide many features to the users to import requirements from source document, define attribute values, filter and display the database contents, export requirements in various formats, define traceability links and connect requirements to items stored in other software development tools [14].

The main reason why every software development project should use a requirements management tool is because it provides automated assistance that help to manage requirements as development progresses. It also helps to perform the following tasks [14]:

- **Manage versions and changes.** Requirements management involve organizing and storing relevant information about requirements, so that the RM tools help to manage history of the changes made to every requirement so that the previous requirements can be reviewed if necessary as well as keeping the updated versions of requirements.
- **Store requirements attribute.** A variety of information, such as attributes about each requirement should be stored in a central repository. Every team member should be able to view the attributes and selected team members should be allowed to update their values. Usually, the requirements management tools generate several system-defined attributes, such as date creation and version number. The tools also allowed the users to define additional attributes of various data types, such as author, person responsible, origin, rationale, release number, status, priority, cost, difficulty, stability and risk.
- **Facilitate impact analysis.** Most of requirements management tool enables requirements tracing by defining links between different types of requirements, between requirements in different subsystems and between individual requirements and related system components. These links help to analyze the impact of a proposing change on a specific requirement by identifying other system elements that might affect. It also provides trace ability each functional requirement back to its origin.
- **Track requirements status.** Requirements management tool has a central database to store all the requirements

The use of requirements management tools has become essential by considering the size and complexity of



development efforts [16]. However, a study [12] of a survey in the Malaysian software industry revealed that there is no appropriate approach of managing requirements. In addition, it is also reported that there is lack of using best requirements management practices among software engineers. Therefore, it is necessary to commence and promote requirements management tools in the Malaysian software industry.

There are some commercial off-the shelf- requirements management tools such as DOORS [17] and Rational Requisite Pro [18]. However, these tools use different concepts; have different capabilities and differing degrees of maturity with respect to their applicability in system engineering projects [16]. According to [13], 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. It is also reported by [13] that the industry is lack of using sophisticated tools.

As a result, there is a necessity to develop a requirements management tool that is appropriate for Malaysian software industry. In order to guarantee that the required requirements management tool is fit to the Malaysian software industry, it is important to identify the elements of the tool and perform a comparative study among the best available requirements management tools in the global market.

### III. ELEMENTS OF REQUIREMENTS MANAGEMENT TOOL

Developing the elements for RM tool is begun with the literature study. The requirements for the RM tools are presented in a couple of research articles; the sixteen requirements for the inter-organizational IT support (IOIS) for collaborative product development are introduced in [19] as well as a set of requirements for RM tools in the area of automotive and aircraft industry is presented in [16]. In addition, detailed list of characteristics of requirements management tools is introduced in [9] and [20] have defined a requirements list for the RM tools used in software engineering as:

- Maintain unique identifiable description of all requirements.
- Classify requirements into logical user-defined groups.
- Specify requirements with textual, graphical, and model based description.
- Define traceable associations between requirements.
- Verify the assignments of user requirements to technical design specifications.
- Maintain an audit trail of changes, archive baseline versions, and engage a mechanism to authenticate and approve change requests.
- Support secure, concurrent co-operative work between members of a multidisciplinary development team.
- Support standard systems modeling techniques and notations.

- Maintain a comprehensive data dictionary of all project components and requirements in a shared repository.
- Generate predefined and ad hoc reports.
- Generate documents that comply with standard industrial templates.
- Connect seamlessly with other tools and systems.

Thus, the elements for RM tool are collected from literature study and market survey. In the Market survey, a set of questionnaire is used to gather what are the elements that the software engineers wish to have as describe in chapter 3. By combining the information from literature study and market survey, the preliminary for RM tool were defined. This preliminary criterion is divided into general element and specific criteria. The general element is the general features that the software tool should have, whereas the specific element is requirements that are specific for the tool.

#### A. General elements

The general elements are important because it describes the features that the tool should accomplish in order to fit the software industry needs. The Table I below presents the general elements for RM tools and follows by detailed explanation.

TABLE I. THE GENERAL ELEMENTS

Elements	Description
Usability, simplicity and customization	The tool should be easy to use. Not too much training and administration needed. The tool should not create additional tasks and deployment should not require extensive customization.
Access control	The tool must have tight access control whereby each participant has appropriate access to the data. (Role-based, project-based and task based access control.)
Tailoring and Extensibility	The tool must be adaptable and extensible to the needs of the organization or project.
Free licensing and full version availability	The tool should be free licensing that allows the user to use the tool in full version without limitation.
Database centric	The tool should be database centric, but also support document management.

**Usability** is an obvious need for a tool supporting collaborative way of working. In order for companies to take tools in use, the tool should not create additional tasks and complicate the development work. In addition, **simplicity** (e.g. training and administration) and ability to operate without extensive tool **customization** are important factors, especially for small companies.

**Access control** is important in collaborative development environment since, for instance, persons from external organisation should not see all proprietary information in the company's data systems. Moreover, it is not necessary for developers to see the project budget, for example, and a Quality Assurance (QA) person can only read requirements, editing is not possible for them. The tool should support restricting a particular user group's access to certain

information and, in general, control accessing to the tool by passwords and data protections.

**Tailoring and extensibility** is practical when the company has many projects of different sizes, and many different tools are used with the RM tool. The tool must be easily adaptable and extensible to the needs of the organization or project.

**Free licensing and full version availability** is the important feature that could promote the user to use the tool as it is free and available in full version.

**Database centric** means that the tool is a database centric tool that concentrates on keeping all the information into a database. However, the RM tool should also be able to manage and generate documents. It is important that the tool should ensure that information containing in the database should be similarly in the documents.

### B. Specific Elements

The specific element is defined in the Table II below and follows by detail explanation.

TABLE II. THE SPECIFIC ELEMENTS

Elements	Description
Requirements identification	The tool should support the identification of requirements. The requirements ID, which is a number for each individual requirement is mandatory.
Requirements classifying and viewing	The tool must be able to classify requirements into logical user defined groups.
Requirements baselining	The tool should be able to manage functional and non-functional requirements that the development team has committed to implement in a specific release.
Change control	The tool must : Offer a possibility of handling formal change requests. All changes to the requirements must be tracked and kept in the database. The tool should be able to update the requirements document.
Version control	The tool should be able to identify: Requirements document versions Individual requirements versions
Status tracking	The tool have to : Define possible requirement statuses Record the status of each requirement Reporting the status distribution of all requirements.
Requirements tracing	The tool ought to : Define links to other requirements Define links to other system elements
Use case specification generation	The tool must be able to generate use case specifications documents. The tool uses predefined document definitions to generate documents with current data from the database.
List of requirements generation	The tool should be able to generate a list of requirements as a support documents.
Requirements linking to system elements	The tool should be able to keep functional requirements, the design components and code modules that address each requirement, and the test cases that verify its correct implementation.
Authentication procedure	The tool should allow the different person with different roles to log in to the tool. The tool

	should restricted it's functions to the different users.
Project definition	The tool should allow a project to be defined in order to keep requirements separately from other projects.
Create user	The tool should be able to create user id and password with different roles. This is important for the user to login and use the tool efficiently.

**Requirements identification** means the ability to identify every single requirement so that distinguishing them from each other is easy. This can be done with requirement identification numbers and with the help of requirements attributes. In addition, the tool must support requirements prioritization, because some requirements are more important than others are, and features that are more important have to be implemented first.

**Requirements classifying and viewing** is the ability to classify requirements into logical user-defined groups, thereby offering different views of the same data to the different users. A view offers the possibility to view and change a freely defined collection of parts of the data of several projects in a freely configurable representation.

**Requirements baselining** is the ability to maintain and manage the set of functional and non functional requirements that the development team has committed to implement in a specific release.

**Change control** is the most important feature in the requirement management tool. The tool must provide a possibility of tracking all changes to the requirements and keep them in the database. The history of the requirements changes (who, what, when, where, why, how) needs to be registered. Change control allows to track the status all proposed changes, and it helps ensure that suggested changes aren't lost or overlooked.

**Version Control** is the ability to control versions of both individual requirements and requirements documents.

**Status tracking** is the ability to track the status of the requirements in the baseline. Requirements tracing is the ability to manage the logical links between individual requirements and other project work products.

**Use case specification generation** means that the desired requirements are gathered from the database to the requirements document. It is not practical to print the whole contents of the database to the document, but only the appropriate requirements. The tool allows generating use case specifications that follow the industrial standard.

**List of requirements generation** is one of the tool facilities to generate the requirements that have been agreed to implement in the current baseline. This document describes the requirements description with requirements version number.



**Requirements linking to system elements** are the tool's ability to keep requirements and its corresponding design elements, code modules as well as test cases. The system elements will be assigned to each requirement after the work has been completed. Thus, the tool records the requirements with its particular completed system elements.

**Authentication procedure** is the tool's function that allows the different users to login. The tool provides the function based on the user role's whenever they login.

**Project definition** is a tool capability to keep the project's itinerary and identification. This is important in order to ensure all the requirements are kept based on the project identification.

**Create user** is the tool ability to allow different users with different roles to create their username and password. This is important in order to ensure the tool's reliability and performance.

#### IV. COMPARATIVE STUDY

The sales of requirements management tools have been growing steadily in recent years [21]. 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 [20]. 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. Hence, a comparative study is conducted in order to search for any requirements management tools that are suitable for the Malaysian software needs based on the defined elements.

For this study, a set of requirements management tools was chosen. The tools introduced in this chapter are taken into account based on how the tool vendors promise to support various sizes of software engineering projects. The number of RM tools is huge, but this study did not include all of these. The selection elements for tools included were that they are well known and broadly used in industry. In addition, information on these tools is quite well available in the literature and in the Internet. The following list introduces the tools chosen for the study.

- Borland CaliberRM
- Insoft Prosareq
- IBM Rational RequisitePro
- ViewSet PACE
- Igatech Systems RDT
- SpeedDEV RM
- RBC RMTrack
- Telelogic DOORS
- Serena RTM
- Teledyne Brown XTie-RT

All of these tools are analyzed against the defined elements for general and specific criteria. Table III illustrates the summary of the results. It should be noted, that the first criterion in the table (Usability, Simplicity and Customization) is left blank, because evaluating usability should demand a test use with every tool.

As can be seen in the Table III, access control is a well-supported feature by every tool. On the other hand, several tools have tailoring and extensibility features as well as some of them are database centric. However, none of the tools provides free licensing with full version availability.

TABLE III. SUMMARY OF PRELIMINARY STUDY – GENERAL ELEMENTS

General Elements	Caliber RM	PACE	Prosareq	RequisitePro	Systems RDT	RMTrack	Serena RTM	SpeedDEV RM	DOORS	XTie-RT
Usability, simplicity and customization										
Access control	√	√	√	√	√	√	√	√	√	√
Tailoring and Extensibility	√	√	?	?	X	?	√	?	√	√
Free licensing and full version availability	X	X	X	X	X	X	X	X	X	X
Database centric	√	?	√	X	√	X	√	?	√	√

From Table IV, it can concluded that almost all tools fully supported requirements identification, requirements classifying and viewing, requirements baselining, requirements traceability, version control, requirements tracking and change control. While, several tools partially supported the generation of use case and list of requirements, requirements linking to system elements, authentication procedure, project definition and create user. As a conclusion, **there is not any tool that has all the general elements and fully supported specific elements**. Thus, from the chosen requirements management tools, there is no tool that is suitable to assist the software practitioners in developing quality software.

TABLE IV. SUMMARY OF PRELIMINARY STUDY – SPECIFIC ELEMENTS

Specific Elements	Caliber RM	PACE	Prosareq	RequisitePro	Systems RDT	RMTrack	Serena RTM	SpeedDEV RM	DOORS	XTie-RT
Requirements identification	√	√	√	√	√	√	√	√	√	√
Requirements	√	√	√	√	√	√	√	√	√	√

classifying and viewing										
Requirements baselining	√	√	√	P	√	√	√	√	√	√
Change Control	√	√	√	√	√	√	√	√	√	√
Version Control	√	√	√	√	√	√	√	√	√	√
Status Tracking	√	√	√	√	√	X	√	√	√	√
Requirement Tracing	√	√	√	√	√	√	√	√	√	√
Use Case Specification generation	√	P	P	P	√	√	X	?	P	?
List of requirements generation	√	P	P	P	√	√	X	?	P	?
Requirements linking to system elements	√	X	?	√	√	X	√	?	√	√
Authentication procedure	√	√	√	P	√	X	√	?	√	√
Project definition	√	√	√	X	X	√	√	?	√	?
Create user	√	√	√	X	√	X	√	?	√	√

Table legend: √-FULLY SUPPORTED, X-NOT SUPPORTED, P-PARTIALY SUPPORTED, ?-NOT KNOWN

In this paper, usability, simplicity and customization are considered elements that are difficult to evaluate without test use. Due to lack of time and resources, these elements were not evaluated in this thesis. It should be noticed that these results are not comprehensive, but it is only indicative, because assessment data is gathered from vendors' web pages and other available evaluation report and white papers. However, results give some kind of a review to the RM tool support for requirements management nowadays.

## V. CONCLUSION

As a key process of software engineering, requirements engineering plays a crucial role throughout the whole software engineering lifecycle. A lot of research has shown that failures of software projects are often related to poor requirements. Well-defined requirements will increase the likelihood of the overall success of the software project. However, it will not be possible to develop better quality requirements without a having a good practices of requirements management. Since requirements engineering is the starting point of software engineering and later stages of software development rely heavily on the quality of requirements, there is a good reason to pay close attention to automating the requirements management process. It is important to have a tool to support and provide effective management of requirements throughout the software development life cycle.

This paper has defined a set of elements for requirements management tool. These elements, which are general and specific elements, are considered as criteria in comparative study in order to perform a requirements management tool

survey. The results conclude that there is no single requirements management tool that has all elements in order to be employed in the Malaysian market. Therefore, it is a need to develop a requirement management tool that is suitable for the companies in the Malaysian market.

## REFERENCES

- [1] Chaos, A Recipe for Success, the Standish Group International, Inc., 1999.
- [2] MN, Microsoft Process Improvements, Software Industry In USA Will Continue Grow Sources: Software: Global Update, Dec. 9, 2004, <http://www.marketnewzealand.com/MNZ/news/story/13572/1783.aspx>
- [3] P. Naur and B. Randell, Software Engineering: A Report on a Conference, sponsored by the NATO Science Committee, NATO, 1969.
- [4] Extreme Chaos, the Standish Group International, Inc., 2001.
- [5] B. Stephen, "FAA Shifts Focus to Sealed-Back DSR," IEEE Software, March 1996, pp. 110.
- [6] F. Brooks, "No Silver Bullet – Essence and Accident in Software Engineering", IEEE Computer, Vol. 20, No.4, pp.10-19, 1987.
- [7] K. E.Emam and A. Birk, "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability", IEEE Transactions on Software Engineering, Vol.26, No. 6, 2000.
- [8] R. R. Young, The Requirements Engineering. Boston, Artech House, 2004.
- [9] I. Sommerville and P. Sawyer, Requirements Engineering : A Good Practice Guide. New York: Wiley, 1997.
- [10] D. Damian, D. Zowghi, L. Vaidyanathasamy and Y. Pal, "An Industrial Case Study of Immediate Benefits Of Requirements Engineering Process Improvement at the Australian Center for Unisys Software", Journal of Empirical Software Engineering, Vol. 9, No. 1-2, pp. 45 – 75, 2003.
- [11] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques. John Wiley & Sons Ltd, 1998.
- [12] A.Zainol and S. Mansoor, "Investigation into Requirements Management Practices in the Malaysian Software Industry", Proc of International Conference on Computer Science and Software Engineering, Wuhan, China, pp. 292-295, 2008.
- [13] A.Zainol and S. Mansoor, "A Survey of Software Engineering Practice in the Software Industry", Proc of IASTED International Conference on Software Engineering, Innsbruck, Austria, 2009.
- [14] K. E. Wiegers, Software requirements. Redmond Washington : Microsoft Press, 2003
- [15] D. G. Firesmith, "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them," Journal of Object Technology, Vol. 6, Number 1, pp. 17-33, January-February 2007.
- [16] M. Hoffmann, N. Kühn, M. Weber and M. Bittner, " Requirements for Requirements Management Tools", Preceedings of 12th IEEE International Requirements Engineering Conference (RE'04), pp. 301-308, 2004.
- [17] Telelogic DOORS, [www.telelogic.com](http://www.telelogic.com)
- [18] Rational Requisite Pro, [www.rational.com](http://www.rational.com)
- [19] A. Ö. Rönnbäck, Interorganizational IT Support for Collaborative Product Development. Dissertation from the International Graduate School of Management and Industrial Engineering, IMIE. No. 59, Doctoral Dissertation, 2002
- [20] M. Lang and J. Duggan, "A Tool to Support Collaborative Software Requirements Management", Requirements Engineering, 6, pp.161-172, 2001.
- [21] Standish Group International, Special COMPASS report on requirements management tool, West Yarmouth,MA, USA, 1998.