A REQUIREMENTS MANAGEMENT TOOL FOR SMALL AND MEDIUM PROJECTS: AN INDUSTRIAL CASE STUDY

AZIDA ZAINOL
School of Computing, College of Arts and Sciences, Universiti Utara Malaysia

SA'AD MANSOOR
School of Computer Science, Bangor University Dean Street, LL57 1UT Bangor, Gwynedd, United Kingdom

ABSTRACT

In the previous years, there are many requirements management tools available in the market. However, these tools are expensive, complicated, difficult to learn and too sophisticated for small and medium projects. This paper presents a case study of implementing a Requirements Management Tool for small and medium projects (RMT-SMP) in the real industry in order for the RMT-SMP to be feasible for small and medium projects. The case study carried out by defining the hypothesis, selection of the pilot study, identification of different methods of comparison, consideration of the effects of cofounding factors, planning of the case study, monitoring the case study as well as result analysis and report generation. The results have shown that the RMT-SMP is practical and feasible for the small and medium projects in the Malaysian software industry and encourages the practitioners to have a better approach in managing their requirements during software development projects.

Keywords: *Requirements Management, Requirements Management Tool, Requirements Management Practices*

INTRODUCTION

As one of the processes in software engineering, requirement engineering (RE) plays a vital role to ensure the overall success of software projects. Theoretically, 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. In order to ensure the quality of a 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 best practices, techniques and methodologies (Emam, & Birk, 2000); Young, 2004); Sommervile, & Sawyer, 1997), Damian., Zowghi., Vaidyanathasamy., & Pal, 2003).

Requirements have a tendency to change during system development and these changes must be managed. Usually, during the RE process involves a large amount of data and unstable requirements. Thus, RM tools have been developed to help in managing them (Kotonya, & Sommervile, 1998). 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.

The sales of requirements management tools have been growing steadily in recent years (The Standish Group International, 1998). 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 (Lang, & Dunggan, 2001). However, not all of these tools in the market are focused solely on requirements management activities. This is supported by a finding of a survey of requirements engineering tools by (Carrillo, et. al (2011) 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 (Hammer, & Huffman, 1998). However, a study (Zainol, & Mansoor, 2008) 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 a 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 [Hoffmann, et. al (2004). In addition, there is an open requirements management tool developed by Etish software development team on the Eclipse platform, known as 'Useme' (Otis project, 2012). 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 (Zainol, and Mansoor, 2009) only 12.2% out of 74 respondents used requirements management tools, which is 10.8%, claimed used Rational Requisite Pro and 1.4% used another 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 (Zainol, and Mansoor, 2011) that the industry is the 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 a 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. Thus, in this paper the research question is using best practices of requirements management that is applied into RMT-SMP tool rather than using ad-hoc requirements management practices has a positive impact on encouraging the RE practitioners to have a better approach for managing requirements in developing the small and medium software projects.

The intention of this paper is to explore in the real-life context whether the RMT-SMP provides advantages to the practitioners by conducting an industrial case study. However, it is widely believed in the software engineering domain that real-life case studies are suitable for an industrial evaluation of software engineering techniques and tools if they are organized and conducted in a sound way (Wohlin, et.al. 2004).

The main objective of this case study is to investigate how the requirements management tool could offer a better way of managing requirements for small and medium projects in a real industry setting. The scope of this case study is limited to small and medium projects in Malaysian software companies. The results obtained from this case study cannot necessarily be generalized to represent the Malaysian software industry and cannot guarantee that similar success would be achieved in other applications because the data collected in the case study only

from two software development projects. However, in this case study, the requirements management tool was able to promote a better way of practicing requirements management, it is practical and feasible for small and medium projects as well as lead toward developing quality software within the allocated budget to deliver it at the right time. Thus, it is a hope that by having RMT-SMP it can fill the gap in the vast area of software engineering in the Malaysian software industry. Furthermore, the RMT-SMP is able to encourage the practitioners to have a better approach for managing their requirements during software development projects.

ELEMENTS OF RMT-SMP

The elements of a RMT-SMP have been determined extensively and it is widely describe in (Kitchenham & Pickard, 1995). It is important to identify the elements that a RMT-SMP should have in order to be feasible for the small and medium projects in the Malaysian software industry. Thus, the elements are divided into general and specific elements.

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 1 below presents the general elements for RM tools and is followed by a detailed explanation.

Table 1

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.

Specific elements. The specific element is defined in the Table 2 below.

Table 2

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 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 has 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

(continued)

Elements	Description
Use case specification generation	The tool must be able to generate use case specification 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 document.
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 restrict its functions to the different users.
Project definition	The tool should allow a project to be defined in order to keep requirements separate 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 log in and use the tool efficiently.

METHODOLOGY

Generally, the case study carried out the following steps (Kitchenham & Pickard, 1995): Definition of hypothesis, selection of the pilot study, identify the method of comparison, consideration of the effects of cofounding factors, planning of the case study, monitoring the case study as well as result analysis and report generation.

DEFINITION OF THE HYPOTHESIS

The hypothesis of the case study is defined as:

Using the best practices of requirements management that is applied into RMT-SMP tool rather than using ad-hoc requirements management practices has a positive impact on encouraging the RE practitioners to have a better approach for managing requirements in developing the small and medium software projects.

The hypothesis can be considered true if the following items are met:

1. The RE practitioners themselves indicate that the RMT-SMP has a positive influence on the requirements management practices when compared to ad-hoc requirements management practices.
2. The RMT-SMP is considered suitable for the given Malaysian software projects when the tool's features can meet the general and specific elements.

In order to examine the validity of the hypothesis, the metrics shown in Table 3 have to be used. There are a lot of other factors that interplay with each other to contribute to the success of requirements management, such as the knowledge of RE practitioners and management commitment. Moreover, there are also a lot of variables that are required to be measured and controlled during the case study. However, in this case study, the focus is on the positive and negative effects that will bring the success of requirements management. This is important in order to identify the merits or problems based on the empirical evaluation under the context of the case study, since it is beyond the scope of the research to investigate the causal relationship of all factors interacting in the case study as well as in the research.

It is a hope that RMT-SMP would provide many advantages and help for the development of software projects. Thus, as long as the RMT-SMP is capable of handling requirements management and is shown to be feasible as well as helpful to the software projects, it will indicate the overall feasibility and merit of the RMT-SMP.

Table 3

Variables to be Measured

No	Variables to be measured	Notes
1	Total number of (atomic) requirements in the final requirements specification	Atomic requirements are defined as lower level requirements with one specific function and cannot be further broken down into a lower function (Salzer, H. 1999).
2	Number of analysts involved	Analyst plays the role of requirements engineers as well
3	Number of developers involved	Analyst can also be a developer or tester when they are required by the project.
4	Number of original requirements	

(continued)

No	Variables to be measured	Notes
5	Number of requirements deleted	
6	Number of requirements rejected	
7	Number of change request	
8	Number of requirements change approved	
9	Number of requirements change rejected	
10	Number of requirements change evaluated	
11	Number of requirements change verified	
12	Number of requirements change modified	
13	Number of requirements change completed	
14	Number of completed change request	
15	Project duration	Include the planned duration and actual duration
16	Effort in person-month	Can be calculated from variable 3 and 15
17	Cost overrun in terms of the Effort in person-month	Can be calculated from variable 3, 15 and 16
18	Software project budget	
19	The number of software product quality expectation	

SELECTION OF THE PILOT STUDY

In this case study, the aim is to evaluate the practicality of the RM-SMP to manage requirements in small and medium projects. Thus, it is necessity to identify two software development projects to represent small and medium projects. A project is considered as small when overall atomic requirements are less than 500 and medium project when the number of atomic requirements is between 500-1000. Based on these conditions and considerations, a project called E-Filing is identified at company Z (the name of the company is withheld for reason of private and confidential). Company Z is a semi-government agency and the case study is conducted at their Information Technology Department. As the development of

E-filing involved 250 requirements, this is considered as small projects. Another project is identified in Company Y and it is known as Human Resource Management System (HRMS). The number of requirements is 650 and it is consider as medium project.

IDENTIFY THE METHOD OF COMPARISON

In this step, it is important to identify a small project similar to E-filling project and a medium project similar to HRMS. In addition, the small and medium projects should have the similar characteristics as E-filling and HRMS; and were previously carried out in company Z and Y. Thus, in company Z, a previously small project is identified as a Project Management and Monitoring System (PMMS). While in company Y, a prior medium project is known as Office Documents Management System (ODMS). The comparison is conducted for small projects; between E-filling and PMMS and medium projects; between HRMS and ODMS. In PMMS and ODMS, the requirements management practices were ad-hoc and there were no requirements management tool getting involved. On the other hand, in E-filling and HRMS, the requirements management practices are defined and using the RMT-SMP to manage their requirements. The results are significant to show the comparison.

Identify the Method of Comparison, Selection of the Case Study Methods and Considerations of the Confounding Factors

This case study involves the development of E-filling and HRMS projects. The comparisons help to examine the impact of RMT-SMP in managing the requirements in order to guide the practitioner to practice better requirements management activity. However, some factors such as the commitment of management and knowledge of RE practitioners are similar among the projects and therefore, no further investigation is needed.

PLANNING THE CASE STUDY

In planning the case study, the following steps are planned:

1. Define and document the context of the case study (objectives and hypothesis)
2. Define the metrics to be used in the projects

3. Introduce RMT-SMP to the RE practitioners by demonstrating and training them to use the tool
4. Software project development
5. Determine the methodologies for analyzing the quantitative data, and comparison of data
6. Assign the tasks and responsibilities to the people who are involved in the case study.

MONITORING THE CASE STUDY AGAINST THE PLAN

This case study involves the development of two different software projects at different companies. Thus, it is important to monitor the projects' progress and compared the results with the plan. In addition, the author involves directly with the projects in order to ensure the tool is conducted correctly. Although the tool is being introduced and trained with the team members, the author keeps on monitoring the team members when they used it. The data collected during the project development are summarized in the Table 4 for small projects and Table 5 for medium project.

Table 4

Result from E-filling and PMMS

Variables measured	E-Filing	PMMS
Total number of (atomic) requirements in the final requirements specification	250	225
Number of analysts involved	4	4
Number of developers involved	4	4
Number of original requirements	130	120
Number of requirements deleted	25	0*
Number of requirements rejected	20	0**
Number of change request	10	0***
Number of requirements change approved	9	0***
Number of requirements change rejected	1	0***

(continued)

Variables measured		E-Filing	PMMS
Number of requirements change evaluated		9	0***
Number of requirements change verified		9	0***
Number of requirements change modified		9	0***
Number of requirements change completed (installed as work product)		9	0***
Number of completed change request		9	0***
Project duration	Planned	6 months	6 months
	Actual	6 months	9 months
Effort in person-month	Planned	24	24
	Actual	24	36
Cost overrun in terms of the Effort in a person - month	Number	0	12
	% over the total effort of the project	0	50%
Software project budget (RM)	Planned	10,000	15,000
	Actual	9,500	18,000
The number of software product quality expectation	Planned	6	6
	Actual	6	4

Notes:
0* indicates that the no requirement was deleted
0** indicates that requirement rejected was never recorded
0*** indicates that no requirements change management was conducted

Table 5

Result from HRMS and ODMS

Variables measured		HRMS	ODMS
Total number of (atomic) requirements in the final requirements specification		650	680
Number of analysts involved		6	6
Number of developers involved		6	6
Number of original requirements		500	490
Number of requirements deleted		50	0*
Number of requirements rejected		20	0**
Number of change request		17	0***
Number of requirements change approved		15	0***
Number of requirements change rejected		2	0***
Number of requirements change evaluated		15	0***
Number of requirements change verified		15	0***
Number of requirements change modified		15	0***
Number of requirements change completed (installed as work product)		15	0***
Number of completed change request		15	0***
Project duration	Planned	9 months	10 months
	Actual	9 months	18 months
Effort in person-month	Planned	54	60
	Actual	54	108
Cost overrun in terms of the Effort in person-month	Number	0	48
	% over the total effort of the project	0	80%
Software project budget (RM)	Planned	50,000	60,000
	Actual	48,000	75,000
The number of software product quality expectation	Planned	10	10
	Actual	10	7

Notes:
0* indicates that the no requirement was deleted
0** indicates that requirement rejected was never recorded
0*** indicates that no requirements change management was conducted

RESULT ANALYSIS AND FINDINGS

The data collected during the development of E-filing was compared with the previous project, PMMS that did not have proper requirements management practices. Both of these projects have similar project attributes. Table 4 and Fig. 1 present the result of comparison between these projects. It can be seen that, both of the projects are similar in number of analyst and developer as well as having the same project duration. Even though the E-filing has 25% more atomic requirement than PMMS, the E-Filling project able to complete its development as planned, within the budget and met the software quality expectations. Moreover, the E-Filling does not have a cost overrun, as it can develop the project within the time. While the PMMS project has 50% cost overrun. From the budget allocated, it can be seen that E-Filling project managed to be developed within the budget. On the other hand, the PMMS project failed to control the budget, as they accessed the budget approximately 20%. When comparing the software product quality, it's also shown that the E-Filling project managed to meet the quality expectations, while the PMMS project failed to do that.

From Table 4, the PMMS project involved no repository purposely in documenting the rejected documents. Although these were only rejected requirements, they might be useful in the future. Additionally, the PMMS project also did not have the change management activity. So, when there were any changes in the requirements, it was difficult to handle it. This became more difficult, if the changes occurred after the analysis phase. The possible reasons behind these circumstances are that the E-Filling project has a proper practice in managing requirements as well as having a tool to help the team members to conduct requirements management activity. Thus, this can be formal evidence that having the best practices and tool in conducting the requirements management activity lead to delivering software within the budget on time without compromising the software quality expectations.

The results of comparison for medium projects are presented in Table 5 and Fig. 2. Both of the projects have almost similar project attributes and they are different in term of managing their requirements management during software project development. In HRMS project, the atomic requirements are carefully managed by using RMT-SMP while in ODMS project involves ad-hoc requirements management practices. Although both of the projects having the same number of analyst and developer, HRMS managed to deliver software on time, within the budget. The ODHM has 30 atomic requirements and the duration is a month more

than HRMS, but, ODHM was no successfully completed the projects on time. ODMS took about another 8 months to complete it and able to satisfy 7 software product quality expectations out of 10. The cost for developing ODHM is also over run. Fig. 2 shows the number of atomic requirements for HRMS and ODMS. It is clearly seen that ODMS did not compile the deleted requirements.

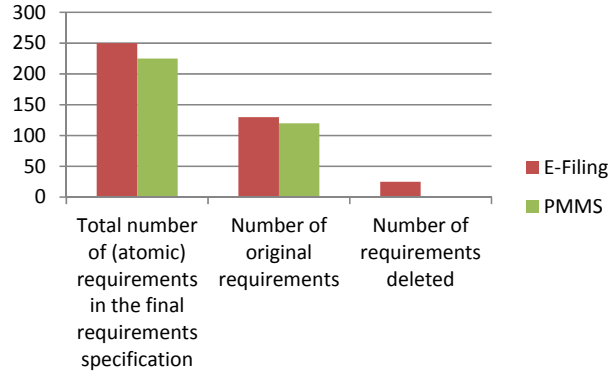

Figure 1. Comparison of variables between E-filing and PMMS.

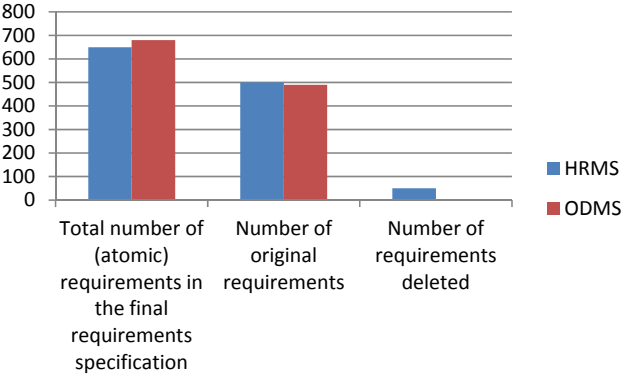

Figure 2. Comparison of variables between HRMS and ODMS.

The HRMS is able to manage all requirements change request and all the changes are going through a well managed procedure. However, in ODMS project, the change requirements request was not cautiously

handled and documented. Thus, it can be concluded that the reason why ODMS fail to deliver software on time, within the budget because ODMS did not have requirements management practices and tool to support it. From both of comparison, it is clearly shown that the RMT-SMP plays a vital role in managing requirements in small and medium projects.

DISCUSSION

The result from quantitative data could be concluded that the practitioners in small and medium projects have improved their requirements management practices compared to during the development of previous project. The following observations were made throughout this case study:

1. The requirements changes were handled carefully and been analyzed, evaluated and modified systematically. This indicates that the RMT-SMP is capable of handling the requirements management especially if there are any requirements changing after the analysis phase.
2. Every requirement identified in the projects was documented and it could be tracked at any time during software development. Even the rejected requirements are documented and able to preview at any time. This shows that there is no lack of requirements abandon and the tool is able to perform as a repository for the references.
3. The project is developed by a team member with four different roles. Although they played different roles, they were able to use the tool based on their needs. Thus, it can be stated that the tool is able to be used in every phase of software development.
4. Requirements management is a part of requirements engineering is not the sole duty of requirements engineers. The involvement of developers and senior management in the process of managing requirements under the leadership of requirements engineers has a positive impact on the project.

The result from this case study is used to compare the RMT-SMP features against the elements of an RMT-SMP. This comparison is essential in order to investigate whether the RMT-SMP could fit the Malaysian software industry. Table 6 and Table 7 below summarizes the result.

The result below is considered valid in this case study only, and it cannot be generalized to represent the Malaysian software industry because this case study only involves two software development projects. However, the results obtained shown that the RMT-SMP features met the general and specific elements. Thus, it can be concluded that, in this case the E-filing project, the RMT-SMP is suitable for the given Malaysian software project.

Table 6

The Comparison of RMT-SMP Features with General Elements

General Elements	RMT-SMP
Usability, simplicity and customization	√
Access control	√
Tailoring and Extensibility	√
Free licensing and full version availability	√
Database centric	√

Table 7

The Comparison of RMT-SMP Features with Specific Elements

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

Specific Elements	**RMT-SMP**
Requirements identification	**√**
Requirements classifying and viewing	**√**
Requirements baselining	**√**
Change Control	**√**
Version Control	**√**
Status Tracking	**√**
Requirement Tracing	**√**
Use Case Specification generation	√

(continued)

Specific Elements	RMT-SMP
List of requirements generation	√
Requirements linking to system elements	√
Authentication procedure	√
Project definition	√
Create user	√

The case study presented in this paper could be an example to indicate that RMT-SMP provided benefits and helps for the small project as in E-filling and medium projects as in HRMS. However, the result from this case study cannot be used as formal evidence that the RMT-SMP, which is a requirements management tool, will always provide the best solution for managing requirements and the success of software project development. Thus, the following factors that reduce the validity of the case study have been recognized:

1. **Management commitment**
   The management of two projects had different levels of commitment to the requirements management process. Management in the E-filling and HRMS projects gave support for using requirements management practices during software development. However, this is not considered as a major effect on the good result in this case study
2. **Learning effects and training**
   The learning effect plays a role because both of the projects are in the same domain application. However, since the PMMS and ODMS projects were a previous project and not conducted at the same time as an E-filling and HRMS projects, the learning effect should not be considered as a major reason that lead to the success in this case study.
3. **Other factors**
   The factors related to the personal attitudes and experiences have influenced in using the RMT-SMP. However, this is only a minor effect toward the success of this case study.

On the other hand, based on this case study, it is likely to state that The RMT-SMP is practical and feasible for the given Malaysian small and medium software projects when the software project could be delivered on time within the budget and met the quality expectations.

Moreover, the result from this case study supports the fundamental assumption made by the RE community that's getting high-quality requirements as well as documenting it early on will reduce rework and overall cost development.

CONCLUSION

This paper describes a case study of implementing RMT-SMP in the Malaysian software companies from the quantitative analysis perspective. On the other hand, the qualitative analysis is also conducted in order to show the feasibility of RMT-SMP for the small and medium projects in the Malaysian software industry. The result of qualitative analysis will be discussed extensively in another paper.

The result of this case study has shown the success of applying the RMT-SMP during software project development for small and medium projects. Therefore, it can be concluded that the hypothesis defined in the section 2 is true and confirms that:

Using the best practices of requirements management that is applied into RMT-SMP tool rather than using ad-hoc requirements management practices has a positive impact on encouraging the RE practitioners to have a better approach for managing requirements in developing the small and medium software projects.

As a conclusion, the objective of this case study which is to investigate how the requirements management tool could offer a better way of managing requirements for small and medium projects is achieved and relevant to the Malaysian software industry.

REFERENCES

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

Damian, D., Zowghi, D., Vaidyanathasamy, L., & Pal, Y. (2003). An Industrial Case Study of Immediate Benefits Of Requirements Engineering Process Improvement at the Australian Center for Unisys Software. *Journal of Empirical Software Engineering, 9*(1-2), 45-75.

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).

Etish project, (2012). Online Document. URL: http://www.etish.org/useme/about.html (retrieved on 11th April 2012).

Hammer, T., & Huffman, L. (1998). 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 .

Hoffmann, M., Kuhn, N., Weber, M., & Bittner, M. (2004). Requirements for Requirements Management Tools. In: *Preceedings of 12th IEEE International Requirements Engineering Conference*, (pp. 301-3080).

Kitchenham, B., & Pickard, L. (1995). Case Study for Method and Tool Evaluation. *IEEE Software*, 52-62.

Kotonya, G., & Sommervile, I. (1998). *Requirements Engineering: Processes and Techniques*. John Wiley & Sons Ltd.

Lang, M., & Dunggan, J. (2001). A Tool to Support Collaborative Software Requirements Management. *Requirements Engineering, 6*(3), 161-172.

Salzer, H. (1999). ATRs (Atomic Requirements) Used Throughout Development Lifecycle. In: *12th International Software Quality*.

Sommervile, I., & Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide*. Wiley, New York.

The Standish Group International. (1998). *Special COMPASS.* Report on Requirements Management Tool.

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

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

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. IEEE Computer Society .

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

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