Requirement Engineering Practices – an Empirical Study

Amjed Tahir and Rodina Ahmad
Department of Software Engineering
University of Malaya
Kuala Lumpur, Malaysia
amjedtahir@siswa.um.edu.my ; rodina@fsktm.um.edu.my

***Abstract*—Requirement Engineering (RE) phase has been regarded as one of the important phases in the development process. Inadequate engineering of requirements can lead to more expensive errors in the later software development phases. Even though there are many methods and techniques which have been proposed in the literatures, many of these methods and techniques have not been widely practiced in the industry. To be able to rectify the situation, the assessment of the current practice is crucial. The main goal of this work is to investigate the software engineering practices especially the requirements engineering practices in the Malaysian software industry. Many of the practicing software developers are the product of the local educational institutions. The findings may help the industries to plan for enhancements in the requirement engineering practices. This research uses a survey instrument to gather data from 27 Malaysian based software firms. The main contribution of this research is the identification of the most practiced requirements engineering activities and the least practiced requirements engineering activities in the software firms.**

Keywords- Requirement Engineering; Empirical Study; Software Engineering Practices

I. INTRODUCTION

Requirement gathering and analysis appears as one of the key processes in any software development. Requirement engineering (RE) covers several activities which include requirements elicitation, analysis, negotiation, specification and validation. Numerous studies have shown the important role of focusing on requirements engineering process for successful software projects. There are a clear relation between the requirement gathering and analysis and software quality [1]. Requirements engineering problems are universal and, indeed, very significant [2]. A study on requirements errors in safety-critical embedded system showed that 60% of errors in critical systems were the results of requirements errors and failures [3]. Alternatively, in an empirical study on requirements engineering, Hell et al. [4] found that out of a total of 268 development problems cited, almost 50% (128) were requirements problems.

In the last two decades, requirements engineering problems becomes an acceptable factor in the inefficiency and failure of most of the software projects [2]. Many software projects have failed because of the poorly defined set of requirements [5]. Based on all the reported studies, clearly significant benefits can accrue from improving the quality of requirements and, by implication, requirements engineering processes [2].

The importance of the RE lies in the importance of the product itself. Improving RE process have the potential to reduce the development costs and time and increase the quality of the software system [5].Thus, In order to develop better RE practices, it is critical to assess and further understand the current practices and problems faced by the software practitioners. .

Malaysia is one the fastest development and fast growth economics countries in the South East Asia region (one of the main areas of outsourcing [6]). It's important to note that most of the Malaysian software development firms are small, which matched the fact stated by Fayad et al. [7]. As mentioned in [6], the majority of the software development firms are small due to the fast and flexible development, though; they develop significant products that need effective software engineering practices tailored to their size and type of business.

This work intends to assess the current practices within Malaysian software development firms and, furthermore, to see which RE practices are applied the most. The empirical study targets small and medium size firms. In general, this paper presents the result of a survey based on data collection from 27 Malaysian software firms, in an attempt to further understand the current practices and main development activities.

The rest of this paper is structured as the following: section 2 presents some similar and related works. Section 3 discusses the research methodology and the data collection approach. Section 4 includes the analysis part of the data then it followed by section 5 where a discussion about the study is presented. Finally, section 6 includes the conclusion and our future work.

II. RELATED WORK

A few surveys have been conducted in the area of RE and software engineering practices.

Hall et al. [4] discuss some problems in 12 companies that they surveyed. Based on their findings, they found that requirement process (gathering and analysis) is the key problem in the software development process. On the other hand, [8] have done a field study on RE practices in the Australian software development industry. The study involved 28 software projects in 16 different software development firms. The study examined, in details, the RE practices and then compared the result of these findings with other well-known RE models in order to identify the gap between theory and practice. In addition, Solemon et al. [9] accomplished a study on general RE problems in 63

978-1-4244-5392-4/10/$26.00 ©2010 IEEE

Malaysian based software firms. Their work presents a survey result of RE problems within these 63 companies. The study concluded that companies with CMM-DEV (capability maturity model for development) certification and others whom do not have any certification have no significant difference in almost all of the problems they discovered. Most of the problems they discovered were requirements-based problems rather than organizational problems.

Niazi and Shastry [1] identified the role of RE in software development process. The work details an empirical study on requirements problems as identified by eleven Australian software companies.

III. Data Collection Techniques

This research follows an empirical approach based on a survey. We are conducting a research based on a real world investigation of software developments firms. This method is highly recommended for software engineering projects especially to understand the current practices or trends [10].

For this study, we carried out two data collection techniques: questionnaires and face-to-face interviews.

The questionnaire is time and cost effective method when easily can collect data from a big number of respondents [11]. Conversely, the interviews are much more interactive. Also, researcher can clarify questions for the respondents.

For the questionnaire, it has been designed based on list of questions that were found in [12]. After we adapted the questions from [12], the questionnaire has been designed , created and then tested in order to evaluate its suitability and reliability in terms of language use and comprehension. The pilot testing has been done by sending the questionnaire to a quality assurance manager and two software engineers. Based on the received feedback, a revised version of the questionnaire has been developed and then sent to the participants.

Besides the questionnaires, several interviews were conducted with several personnel from different Malaysian software firms. Our interviews provided detailed descriptions of software development practices in software firms to help identify high-leverage factors for improving such processes. These interviews involved people from different positions of the software development side. However, we conducted our interviews alongside the questionnaires. The interviews were used to assure the quality of the answers gathered from the questionnaires and give additional clarification to the original questions. Though, several in-depth questions were also asked during the interview that could not be asked in the questionnaire due to some privacy policies that followed by some firms. However, it is important to note that we are using a semi-structured interview technique where the interview generally follows more of conversational flow

It is important to note that this paper presents anonymous individual results, as we were trying to get access to some sensitive data about individuals and organizations.

The survey was conducted in a quantitative manner using a questionnaire as the main instrument[13] . However, we perform several interviews with the participants to cross-check some of the information gathered. The questionnaires were distributed to more than 70 firms. The selection of the firms was based on: business focus, company profile and previous collaboration. Among these 70 firms, we managed to get response from 27 firms. These questionnaires were sent by email and we followed that with some phone calls and reminding notes. For some cases, we handed the questionnaire directly to the participants.

A random sample of more than 200 managers, project managers and software engineers were contacted by emails and telephone to request participation in this study. However, we received 27 responses in total. Singer et al. [11] mentioned that the response rate for software engineering research can be relatively low. It has been found that the response rate of software engineering surveys is around 5% [11]. Even though the total number of firms participating is low and may not represent the total Malaysia software firms, we believe that these firms are fully involved in software development and their responses are reliable.

The analysis of the data gathering is shown in the next section.

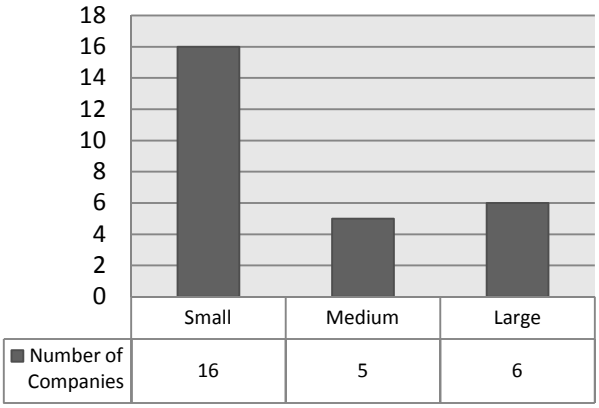

Figure 1: Number and Categories of organizations that participate in our survey

IV. Analysis and Findings

A definition of "small" businesses varies by industry [7]. However, as our focus is on small and medium size of organization. We defined organization's size based on the number of employees. Organization with less than 100 employees are small, Medium (from 101 -200), large (more than 200). Figure 1 shows the numbers and the categories of organizations that we covered in our survey.

The majority of the respondents were software engineers from different levels (juniors, seniors, developers, designers….etc.). Total of thirty three applicants participated in this survey, 27 of them answered the questionnaire. Among these practitioners, 45 % of the respondents were software engineers (Figure 2), 12 %

project managers, 9% managers and 30% others (consultants, team leaders etc). It is important to note that some firms give different names for the same position. As in software engineering, we considered all the jobs that deal with software engineering tasks and functions. For example, some of the participants titled themselves as programmers, web programmers, and software system analyst.

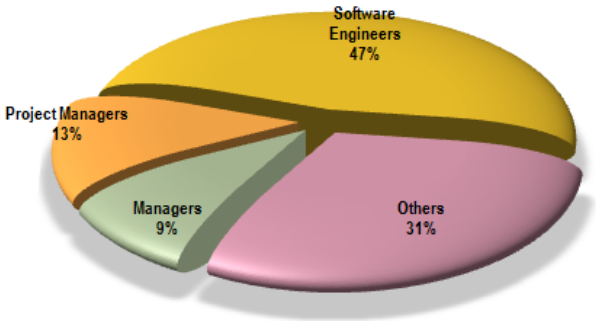

Figure 2: Respondent's background

The total number of people we interviewed was five. All from different firms except that there are two of them who came from the same firm. We have interviewed a project manager, software architect, data-base administrator (DBA) and two software engineers. The result of these interviews has been added up to the results of the questionnaire. During the interview, we basically use the questionnaires as the main drivers for our conversations. However, we did have the chance to ask several more in-depth questions to clarify and get more details from the interviewees.

One of the main questions in the questionnaire is to ask the respondents to allocate 100% of time spent to the core software engineering activities performed in their firm. The question focuses on five main activities of software development which are requirements gathering and analysis, system design, testing and integration, software coding and unit testing and finally software quality assurance. Based on the response, we found that the highest ratio (amount of time) was given to the software coding and unit testing with 32.9% of the total time. It follows by system design (20.6 %), requirement gathering and analysis (19.6 %), testing and integration (19.3 %) and finally (software quality assurance with around 10 % of the total development time and effort (figure 3).

This paper mainly focuses on discussing the present requirements engineering practices. The main aim is to identify the most performed activities and the activities which are not yet become the current practice in the industry. There are essentially eleven main activities within the RE phase that we focus on. We ask the respondents to rank each activity in terms of the frequency of practice.

For the first question, which focuses on conducting a feasibility study, 74% of the firms mentioned that they always or often conduct their own feasibility study before starting any projects. Only 11% of the firms seldom perform feasibility studies on their projects.

The second main activity is the prototyping activity. For misinterpret or poorly understood requirements, only 33% of the firms always or often apply prototyping technique. Less than 7 % always develop prototypes for their requirements. Around 11% percent of the firms do not perform prototyping at all for their project requirements.

Checklist is one of the well-known techniques that is used in the requirements analysis. Thus, 82 % of the firms always or often use checklist for their requirement analysis activity. Only around 7% of the firms seldom use checklist for their requirements analysis.

Similarly, almost 90 % of the firms always or often prioritize their requirement consistently. On the other hand, most of these firms do not use a specific prioritization technique. Only 4% of the firms mentioned the prioritization technique that they use

Nowadays, there are many standard templates or documents that use to describe project requirements. The standard template can help to manage and categorize and further analyze requirements. From the study we found that more than 80% of the participating firms always use standard templates and documents in writing and categorizing their requirements.

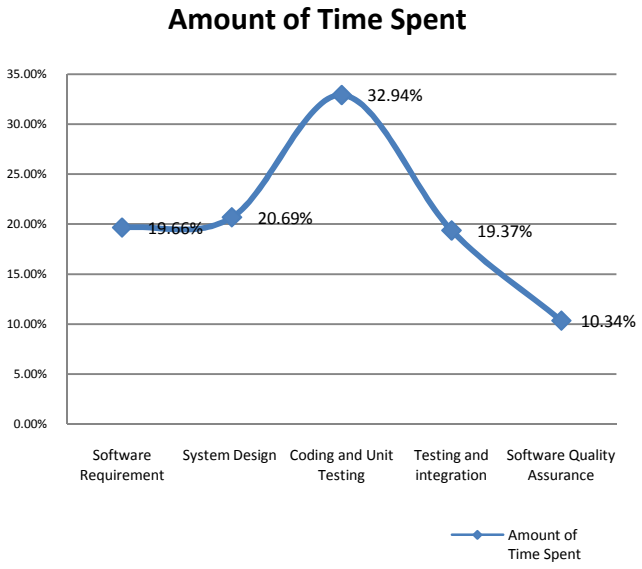

Figure 3: Amount of time per individual tasks

We have asked all the participants to identify whether they use a database or a specific software system for their requirement management. Around 48% of the firms do not use any database or software system to manage projects' requirements. Only 33% of the firms always or often use software or database for managing requirements. While less than 7% sometimes use software system to manage their requirements.

There are a number of factors that may affect the RE process. One of these factors is the reusability. For reusing requirements, 77 % of software firms always or often use requirements that have already been used with some other (similar) systems. Reusing of components can help to reduce time and effort required to perform a specific task. Reusability in requirements can be possibly performed by reusing written requirements (based on previous experiences) from some pre developed, existing systems. Or can be performed in other ways. By providing reusable components, which is a very complicated task and require intensive quality assurance involvement, firms can adopt old, well-implemented components and attach them with the new system in a very manageable manner. This indeed leads to save time, effort and cost of the system under consideration. Additionally, it will reduce errors and bags within the new system as these components normally are almost error free due to the fact that these components have passed many reviews previously before it goes alive, and also the lessons learned from the old system.

In terms of communication, email services have emerged to be a major method of communication. Thus, it's not a surprise that 90 % of the responded firms use email to support requirements negotiations. All the firms that we investigated use emails for most of the times but at different levels; as some firms prefer to use the face-to-face communication.

Classifying requirements will definitely help to improve process of requirements gathering and analysis. The classification can be done in many different ways such as categorizing the requirements into stable and changeable requirements. For the classification technique, total of 67.5 % of the firms classify requirements (in different levels). Nearly 60 % always classify their projects' requirements. A minor number (only 7%) of the firms do not use any classification technique and never classify projects' requirements. Classifying requirements is one of the important management activities to ensure requirements are well understood and can be easily maintained and updated if changes occur.

Risk analysis is a critical activity to identify potential risks and prepare plan to avoid or mitigate the risks. It is a continuous process that should start at the pre initial stage and then persist into the entire project life cycle. Studying software requirements obliges involvement of risks investigation. Risk analysis for requirements focuses on the stability and changeability of the requirements. In this survey we found that only 60 % of the firms perform a risk analysis on the requirements before the implementation of these requirements. 11% of the firms always perform requirements risk analysis. Around 40 % of the firms often perform risk analysis on requirements while another 15 % has not performed any risk analysis for the requirements.

Another important requirements activity is to produce test cases for the requirements. Producing test cases can help the analysts in variety of way including verifying, clarifying and increase understanding on the requirements. From the survey, 68 % for the firms produce test cases for their requirements. Among these firms, there are 26 % firms that always produce requirements test cases for the projects.

V. DISCUSSION

Good requirements engineering practice is known to have a positive impact on software development projects. In this survey, we intend to investigate the current practice of requirements engineering activities with the aim to identify the most practiced activities and the least practiced activities. These understanding will essentially help us to identify further problems that hinder them from practicing and to identify ways in which these activities can be further inculcated into their practicing culture.

Based on the survey findings, it is worth noting that most of the main activities in requirements engineering process such as feasibility study, requirements prioritization, having standard template for requirements, using software systems for managing requirements, identifying non-functional requirements are well practiced by the surveyed firms. More than 50% of the firms always or often practiced these activities as part of the development process. Among the less practiced activities which received the average of below 3.5 (from 5) are prototyping, conflict resolution, using interaction matrices, risk analysis, organizing formal requirements and including stakeholders in requirements validation. The survey shows that the least practice activity is using interaction matrices to find conflicts and overlaps among requirements. On the other hand, the most practiced activity is the prioritization activity. This finding may indicate that the practitioners are performing essential requirements analysis activity such as selection and prioritization. However, they might have to further equip themselves with more analysis techniques like the interaction matrices which can assist them to identify requirements conflicts and overlaps.

Alexander and Stevens [14] recommended that about 5 % of the project effort should go into requirements, not including specification. This might be around 25 % of the total project duration. In this survey, we found that the total amount of time goes to requirement is about 20 % (including specification). This is a 5% lower than the number suggested by other studies. The finding might give us certain indicator that the Malaysian practitioners are still not spending much effort in the RE phase compared to the other phases and the practitioners are spending most of their development time on coding and unit testing. Based on this finding, it is a need to actually create more awareness especially to the project leaders and managers as well as the software practitioners on the importance of requirements engineering phase and the substantial role that it can play to the success of a software project.

VI. CONCLUSION AND FUTURE WORK

RE activities is considered a complex and uncertain process. Hence, it is critical for software firms to have established RE practices in place to ensure that the software products will be developed in a manageable manner. This

study surveys 27 software firms in Malaysia that develop software systems for different purposes. The study is able to identify the main requirements activities which are most practiced and the least practiced activities. It is also able to identify the main phase in which the most time is spent on during the software development activities which is the coding and unit testing phase.

The second indicator from the survey finding is that the Malaysian practitioners are already practicing main requirements activities such as feasibility study, requirements analysis and requirements selection and prioritization. However, the more subtle activities such as conflict resolution, risk analysis and requirement inspections are not so well practiced yet. However, it seems that most of the surveyed firms are aware of these activities and only not be able to practice them more frequently at this point of time. Subsequently, there are still many potential future works that can be performed in this area which include further investigation on the reasons behind the least practiced activities, identification of main support tools and techniques widely used in the industry and types of training given and required by the practitioners.

We believe that, by addressing these RE related issues, Malaysia software firms will be able to improve the efficiency of their development process which subsequently improve the quality of their software product

Our future work will attempt to propose some guidelines or a framework (based on the identified issues) for software firms that can facilitate them to apply RE practices in a more organized manner.

ACKNOWLEDGMENT

Our gratitude goes to all participating software firms for this survey. Thank you very much for your cooperation and support. This work is funded by The University of Malaya under grant number RG020 09ICT.

REFERENCES

[1] M. Niazi and S. Shastry, "Role of Requirements Engineering in Software Development Process: An Empirical Study," in *the 7th International Multi Topic Conference*, 2003.

[2] I. Sommerville and J. Ransom, "An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement," *ACM Transactions on Software Engineering and Methodology,* vol. 14, pp. 85 - 117 2005.

[3] R. Lutz, "Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems," presented at the IEEE International Symposium Requirements Engineering 1993.

[4] T. Hall, S. Beecham and A. Rainer, "Requirements Problems in Twelve Software Companies: An Empirical Analysis," *IEE Proceedings Software* vol. 149, pp. 153 - 160 2002.

[5] K. El-Emam and N. Madhavji, "A Field Study Of Requirements Engineering Practices in Information Systems Development," presented at the The Second IEEE International Symposium on Requirements Engineering, 1995.

[6] T. L. Friedman, *The world is flat : a brief history of the twenty-first century* Camberwell, Vic. : Penguin, 2006.

[7] M. Fayad, M. Laitinen and R. Ward, "Thinking Objectively: Software Engineering in The Small " *Communications of the ACM,* vol. 43 pp. 115 - 118, 2000.

[8] E. Sadraei, A. Aurum, G. Beydoun and B. Paech, "A Field Study of The Requirements Engineering Practice in Australian Software Industry," *Requirements Engineering Journal,* vol. 12, pp. 145-162, 2007.

[9] S. Badariah., S. Shamsul. and A.-A. Abd-Ghani., "Requirements Engineering Problems in 63 Software Companies in Malaysia," presented at the International Symposium on Information Technology, Kuala Lumpur , Malaysia, 2008.

[10] B. Curtis, H. Krasner and N. Iscoe, "A Field Study of The Software Design Process for Large Systems," *Communications of the ACM,* vol. 31, pp. 1268 - 1287 1988.

[11] J. Singer, S. Sim and T. Lethbridge, "Software Engineering Data Collection for Field Studies," in *Guide to Advanced Empirical Software Engineering*, ed London: Springer 2008, pp. 9-34.

[12] S. Anderson and M. Felici, "Requirements Engineering Questionnaire," vol. 1.0, ed. Edinburgh: The University of Edinburgh, 2001, p. 15.

[13] B. Kitchenham and S. Pfleeger, "Personal Opinion Surveys," in *Guide to Advanced Empirical Software Engineering*, ed London: Springer 2008, pp. 63-92.

[14] I. Alexander and R. Stevens, *Writing Better Requirements* Addison-Wesley Professional, 2002.