INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN BUSINESS & SOCIAL SCIENCES

Vol 14, Issue 10, (2024) E-ISSN: 2222-6990

Investigation of the Current Requirements Engineering Practices among Software Developers in Malaysia

Rosmiza Wahida Abdullah, Sabrina Ahmad, Siti Azirah Asmai

Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka

Corresponding Author Email: rosmiza@utem.edu.my

To Link this Article: http://dx.doi.org/10.6007/IJARBSS/v14-i10/23105 DOI:10.6007/IJARBSS/v14-i10/23105

Published Date: 04 October 2024

Abstract

Requirement engineering (RE) is a phase in software development that involves understanding, documenting, and managing the requirements of a software system. RE is a crucial and critical phase to obtain the essential requirements for the to-be-developed system is a reliable and workable system and, in the end to ensure that the software meets the needs and expectations of stakeholders. In Malaysia, software development teams typically follow industry-standard practices for requirement engineering which includes Requirements Elicitation, Requirement Analysis and Documentation, Requirement Validation, Requirement Management, Collaboration and Communication, Requirements Traceability and Agile and Iterative Approaches. This paper presents current RE practices by software development teams in Malaysia. A survey has been conducted to collect data from 12 companies comprises of varies business domain. The respondents were asked about the importance of activities conducted in practices of requirement elicitation, requirement analysis and negotiation, describing requirements, system modelling and requirement validation. At the end of the survey, the respondents were also asked about the indicators for software quality and software productivity. The results from the survey show an overview of how the software development practitioners perceive the importance of each activity in specific RE practices. This is important for the RE researcher to know for us to recommend methods and approaches that can help the practitioners to simplify the process without compromising the quality of the obtained software requirements.

Keywords: Requirements Engineering Practices, Software Development Practitioners, Survey.

Introduction

Requirement Engineering (RE) is the first and the most crucial phase in a software development project. The definition of requirements engineering (or software requirement) based on SWEBOK (Software Engineering Body of Knowledge) v3.0 (Abran et al., 2004) is activities in software engineering consist of elicitation, analysis, specification, and validation of software requirements as well as the management of requirements during the whole life

cycle of the software product. It also explains the term “requirements engineering” to denote the systematic handling of requirements (Abran et al., 2004). The RE process must be very well performed to ensure quality software requirements. A study by Lafta (2015) stated that a poorly engineered requirements process contributes immensely to the failure of software projects. It is also stated that projects that undermine RE suffer or are likely to suffer from failures, challenges, and other risks (Hussain et al., 2016).

Requirements collected must be correct, complete, and concise to ensure the success of the developed software system. To do that, requirement engineers need to specify the stakeholders and to ensure they participate in providing the requirements (Ghanyani et al., 2018). The process is indeed challenging as it needs to gather and translate the imprecise, incomplete needs and wishes of the stakeholders into complete, precise, and formal specifications.

Thus, this study is to investigate the RE activities that have been practiced by the software development practitioners in Malaysia. The ultimate aim of this study is to provide an insight for further exploration and contribution towards the betterment of RE practices amongst the software development practitioners in Malaysia and in the end, able to help in obtaining quality software requirements.

Following the introduction, Section 2 explains the related work of the study. This is followed by Section 3 which elaborates on the survey method. Section 4 presents the survey findings. Section 5 summarizes the survey results and the final section, which is Conclusion and Future Work concludes the study and the potential work that can be explore more from this study.

Related Work

There are several surveys that have been conducted in the areas of RE and software engineering practices. Numerous studies have attempted to analyse the differences among requirements engineering implementations in different environments.

Among the earliest study in this area, Emam and Madhavji (1995), performed a study regarding the issues that can affect RE practices in information system development projects. Their study focused on aspects such as project management capability, management uncertainty, and the selection of capable personnel for the key roles in RE. The study found that organizational and project problems, or non-technical problems, have a direct impact on the requirements engineering process.

Hall et al (2002), performed a survey regarding RE problems in 12 different companies. The researchers found that the process of gathering and analyzing requirements is the key problem in the software development process. The survey showed that RE caused 48% of all discovered problems.

A year after that, Niazi and Shastry (2003), conducted a detailed empirical study of requirements problems identified through interview sessions with 22 practitioners from 11 Australian software companies. The findings were grouped into two categories: problems faced by organizations with a mature requirement process and problems faced by organizations with an immature requirements process. The RE process is considered mature

if it is based on good practices and well-defined methods. They found that requirement growth and continuing changes were the key challenges for the first group, and an unclear and vague initial requirement was the primary challenge for the second group. The authors provided some guidelines for both types of organizations to achieve better requirements management.

Later, Sadraei et al (2007), performed a field study regarding RE practices in the Australian software development industry; their study involved 28 software projects from 16 different software development firms. Their study examined RE practices and then compared the results with well-known RE models to identify the gap between theory and practice. The study found that the RE process was considered one aspect of the entire software process rather than a separate process. It also found that large projects had better documentation awareness and standards. In addition, the number of employees working on a project affects the existence and number of RE responsibility roles in a project.

Then, Solemon et al (2008), studied RE problems in 63 Malaysia-based software firms. Their study concluded that companies with CMM-DEV (Capability Maturity Model for Development) certification had no significant differences from others without any certification for almost all of the problems discovered. Most of the discovered problems were requirements-based rather than organizational problems. Although the work mentioned the best practices for RE, it did not demonstrate any link between these practices and the RE problems that they have found in their study.

Years later, study of requirements engineering and project success conducted by Rasheed et al (2021), also supported the findings by Emam and Madhavji (1995), and recommended that during the early stages, managers of successful RE teams should consult experts to strengthen and validate the team's knowledge base (Rasheed et al., 2021).

Methodology

An expert opinions method was conducted to gather data regarding critical RE practices for software development projects. The opinions from software development project experts were obtained through an online survey. The experts are software development projects practitioners who have been in the industry for 3 years and above. The RE practices investigated were selected based on Sommerville and Sawyer's (1997) RE good practices guidelines. In their work, the authors suggested a list of 66 RE practices that can help practitioners implement a better RE process.

Survey Instrument

In this study, each expert was asked to assess and ranked 40 RE practices into five Likert-scale categories as the following:

1) Very important: A RE practice is referred to as having 'high degree of importance' and it is mandatory.
2) Important: A RE practice is referred to as having 'medium degree of importance' if it is widely followed in the organization's process or policy but is not mandatory.
3) Neutral: A RE practice is referred to as having 'low degree of importance' if it is.
4) Less important: A RE practice is referred to as having 'zero perceived benefits' if it is never or rarely applied to any software development project.

5) Not important: A RE practice is referred to as having no need to be executed because it does not have any benefit to the RE process.

From this list, we have the degree of importance placed on RE practice by experts based on their experience from previous software development projects. We chose a questionnaire as our primary data collection method because questionnaires are best suited to the nature and type of data that we set out to analyze. A similar study performed by Khan et al (2021), used a similar approach. The questionnaire was developed based on Sommerville and Sawyer's RE good practices guidelines.

To ensure the questions we asked in the survey are understood, relevant and do not mislead, we ran a pilot study that involved two researchers and two industry practitioners in the field of software development projects. This study intended to obtain experts' opinion on the validity of the questionnaire content and also to assess the experts' level of understanding, level of knowledge, and level of difficulty in responding and the level of relevance of the item to the subject area (Wan Abu Bakar and Solemon, 2017). We treated these four points by examining the pilot test responses and making changes based on the feedback as follows:

1) Level of understanding: In some cases, the pilot study participants did not understand some of the questions. To increase the level of understanding and to avoid misinterpretation by the respondents, we provided a description for each of the practices.
2) Level of knowledge: This is not a problem because the respondents were the experts in software development projects.
3) Level of difficulty: Our pilot study did not highlight any problems in responding to the questions. Thus, there were issues in this regard.
4) Level of relevance: None of the pilot study participants questioned the relevance of the questions asked. Thus, we did not face any problem regarding the relevance of the questions.

During our pilot study, we ensured that the questionnaire was adequate in terms of its presentation and clarity. This was accomplished by having two researchers review the completed questionnaire and comment on its length, wording, instructions, and format before it was distributed. We also used the pilot study to assess the time commitment required to complete the questionnaire, which we estimated to be around 15-30 minutes. Based on the pilot study participant's feedback, the questionnaire was refined and sent back to the participants. None of the comments from the second review implied a need to change the revised questionnaire.

Data Collection Implementation

This study was primarily conducted through online platforms, which are Facebook and WhatsApp applications to distribute the survey questionnaires and collect them. We decided to reach the respondents online to get faster response and convenience for both parties, the experts and the researcher. We distributed the questionnaire to approximately 18 respondents from 18 different companies. We managed to collect complete questionnaires from 12 respondents, representing 66.7% of respondents who answered the questionnaires.

These 12 respondents are the software development practitioners from 12 companies in Malaysia.

Findings

The questionnaire was designed to gather data regarding the importance of practices from the 5 main groups of practices based on Sommerville's framework, which are requirements elicitation, requirements analysis and negotiation, describing requirements, system modeling, and requirements validation. We also asked the respondents about the indicators of for software quality and software productivity. We would like to know what the most significant indicators for a quality and productive software are according to the software development practitioners in Malaysia. For the analysis of the importance of each RE practice, the occurrence of a scale value (very important, important, neutral, less important or not important) from each questionnaire was counted. The primary data analysis used a frequency analysis technique to identify the most critical RE practices among the surveyed practices.

Respondent Demographics

Figure 1 shows the distribution of our respondents according to their position in their company. The respondents comprise of 25% analyst (3), 25% software engineer (3), 17% systems engineer (2), 17% project manager (2), 8% programmer/developer (1) and 8% consultant (1).

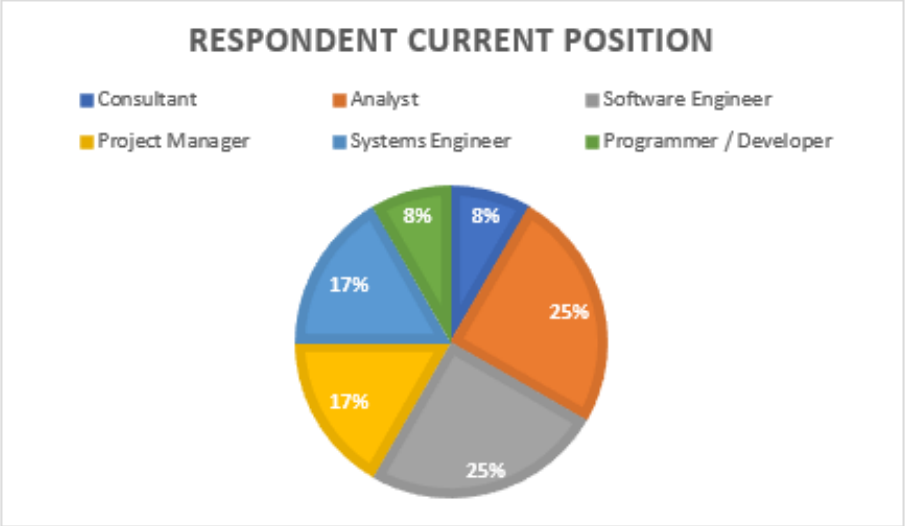


Figure 1: Current position of respondents

Figure 2 shows the respondents' years of experience in software development projects. Most of our respondents, which is 75% (9) of them, have more than 10 years of experience in software development projects. 8% or 1 respondent have 6 to 10 years of experience, and 17% or 2 respondents have 3 to 5 years of experience. Looking at this data, we can say that most of the respondents are experts in software development projects as they have more than 5 years of experience in this domain.

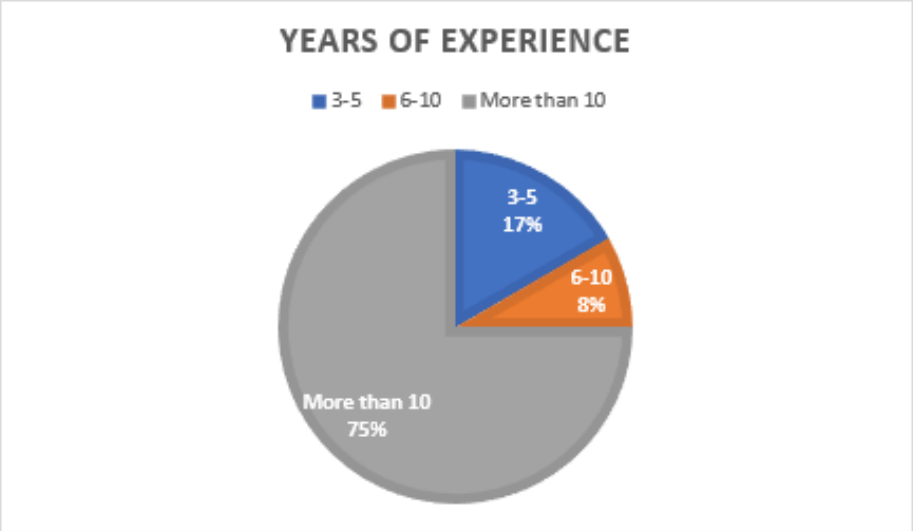


Figure. 2: Years of experience of respondents in software development projects

Figure 3 shows the dimension of the company of the respondents. 67% (8) of them are from the companies that have more than 300 employees and the rest (4) of them are from the companies with 100 to 300 employees.

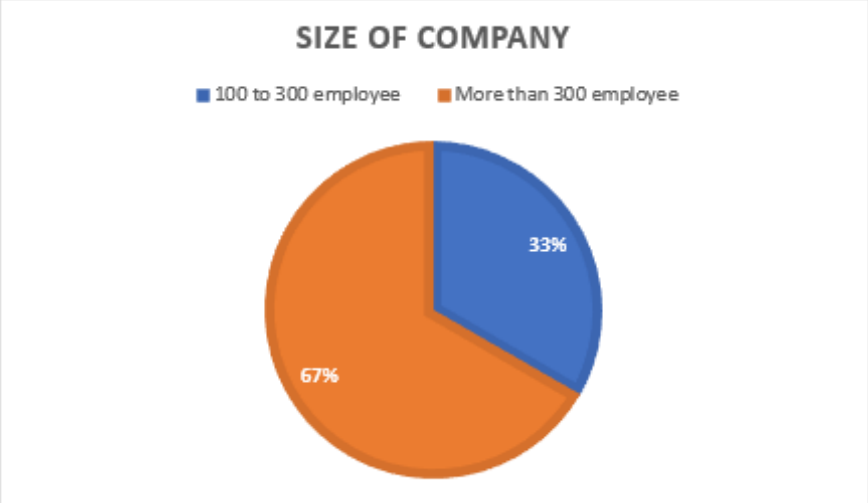


Figure 3: Size of the respondents' companies

Figure 4 depicts the business domain of their companies. Most of the respondents work at information technology (4 respondents) or banking/finance/insurance companies (4 respondents). 17% (2) of them work at government agencies and 1 respondent at energy generation/distribution company and the other 1 respondent works at other domain company.

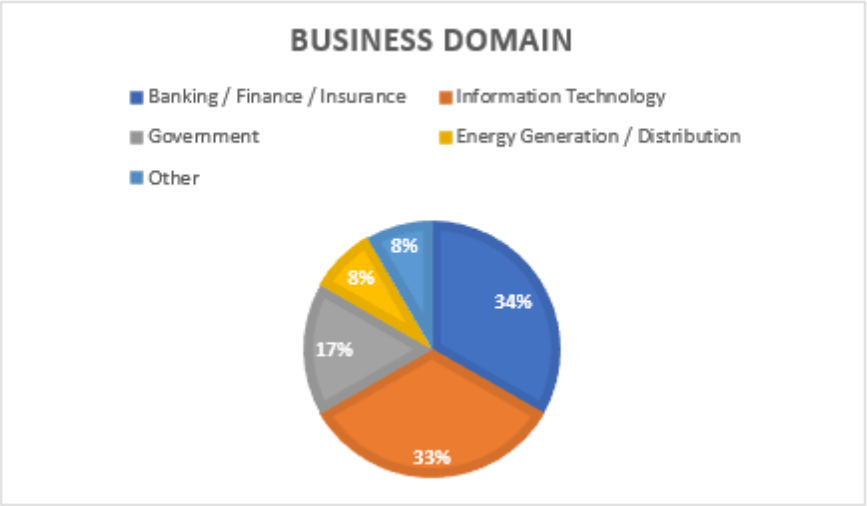


Figure 4: Business domain of the respondents' companies

Concerning the requirement engineering practices, Figure 5 shows that the all 12 respondents selected informal modeling (text and boxes), 11 respondents selected semi-formal modeling (DFD,UML), use cases and storyboarding/whiteboarding, 10 respondents selected scenarios and interviews, 9 respondents selected throw-away prototyping and data mining, 7 respondents selected focus groups, 4 respondents selected quality function deployment (QFD) and finally, 2 respondents selected user stories and formal modeling (Z,VDM,SDL).

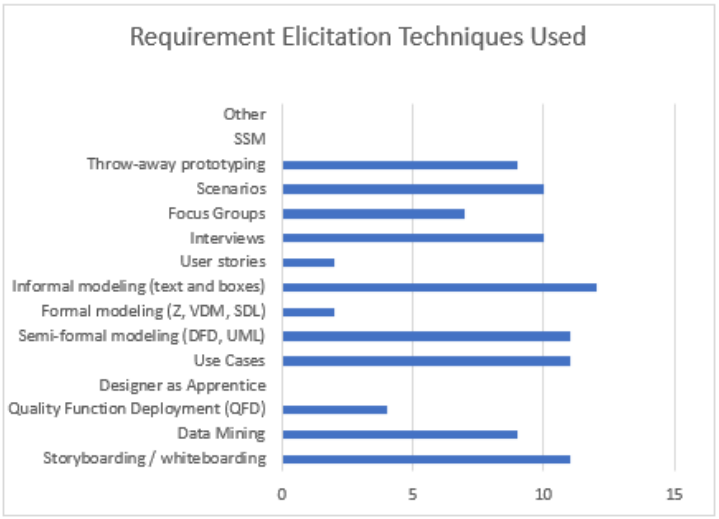
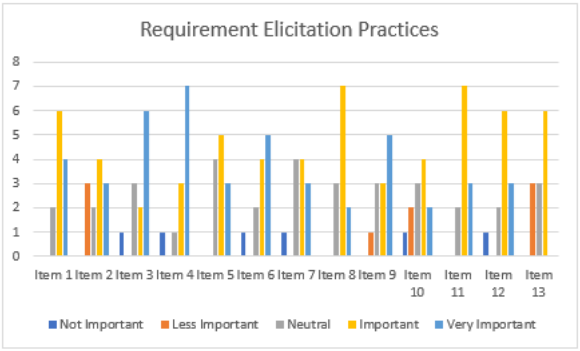


Figure 5: Requirements elicitation techniques used by the respondents

Requirement Elicitation Practices

Regarding Figure 6 shows the assessment result of the requirement elicitation practices. Most of the respondents thought that item 3 (7 respondents) and item 4 (8 respondents) are very important. Half and more respondents also thought that item 1 (6 respondents), item 8 (7 respondents), item 11 (7 respondents), item 12 (6 respondents) and item 13 (6 respondents) are important to be conducted during requirement elicitation process. However, there was 1 respondent thought that item, item, item and item 12 are not important for the requirement elicitation process. Looking at the assessment result for the requirement elicitation practices, which is based on the experts opinion, we conclude that the significant practices to be

conducted during requirement elicitation process are the item 4 (record requirements sources), followed by item 3 (identify and consult system stakeholders), item 8 (record requirements rationale), item 11 (use scenarios to elicit requirements), item 1 (assess system feasibility), item 12 (define operational processes) and finally, item 13 (reuse requirements).

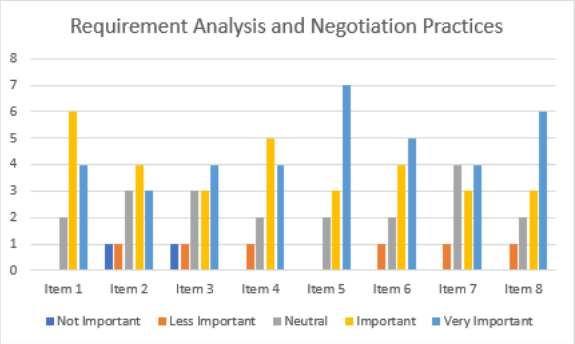


Item	Description
Item 1	Assess System Feasibility
Item 2	Be sensitive to organizational and political consideration
Item 3	Identify and consult system stakeholders
Item 4	Record requirements sources
Item 5	Define the system's operating environment
Item 6	Use business concerns to drive requirements elicitation
Item 7	Look for domain constraints
Item 8	Record requirements rationale
Item 9	Collect requirements from multiple viewpoints
Item 10	Prototype poorly understood requirements
Item 11	Use scenarios to elicit requirements
Item 12	Define operational processes
Item 13	Reuse requirements

Figure 6: Assessment result for Requirement elicitation practices

Requirement Analysis and Negotiation Practices

Figure 7 shows the assessment result for requirement analysis and negotiation practices. Half and more selected item 5 (7 respondents) and item 8 (6 respondents)) as very important practices to be conducted for the requirement analysis and negotiation stage. Most respondents also thought that item 1 (6 respondents) and item 4 (5 respondents) are important practices too. There was also 1 respondent each voted not important for item 2 and item 3. Thus, based on the assessment result for requirement analysis and negotiation practices obtained from the experts' opinion, we conclude that the most important practices for requirement analysis and negotiation are the item 5 (prioritise requirements), item 6 (classify requirements using a multidimensional approach), item 1 (define system boundaries) and item 4 (plan for conflicts and conflict resolution).

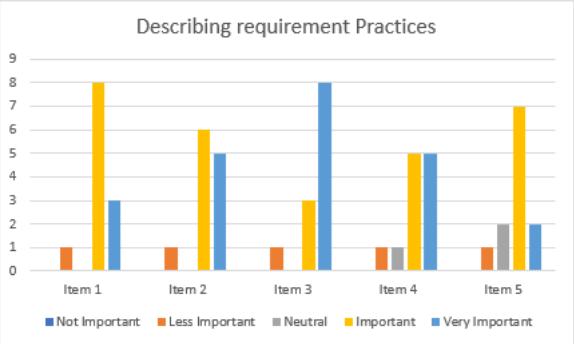


Item	Description
Item 1	Define system boundaries
Item 2	Use checklists for requirements analysis
Item 3	Provide software to support negotiations
Item 4	Plan for conflicts and conflict resolution
Item 5	Prioritise requirements
Item 6	Classify requirements using a multidimensional approach
Item 7	Use interaction matrices to find conflicts and overlaps
Item 8	Assess requirements risks

Figure 7: Assessment result for Requirement analysis and negotiation practices

Describing Requirements Practices

Figure 8 depicts the assessment result for describing requirement practices. Majority of the respondents thought that item 3 is very important practice to do when describing the software requirements. Half or more respondents thought that item 1 (8 respondents), item 5 (7 respondents) and item 2 (6 respondents) are important practices for describing requirements. From the graph, we can see there is no practices has been selected as not important. As such, we conclude that all the 5 describing requirement practices could be done, but the most important are the item 3 (use diagrams appropriately), item 1 (define standard templates for describing requirements), item 5 (specify requirements quantitatively) and item 2 (use languages simply and concisely).



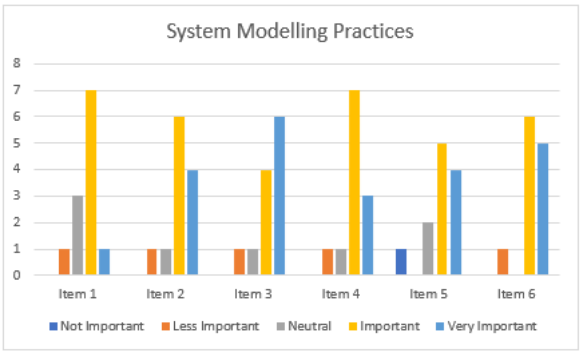
Item	Description
Item 1	Define standard templates for describing requirements
Item 2	Use languages simply and concisely
Item 3	Use diagrams appropriately
Item 4	Supplement natural language with other description of requirement
Item 5	Specify requirements quantitatively

Figure 8: Assessment result for Describing requirement practices

System Modeling Practices

Result for assessment the practices of System modelling is presented in Figure 9. Half of the respondents chose item 3 as very important. Half or more respondents thought that item 1, item 4, item 2 and item 6 are important practices to be carried out for system modelling. Thus, based on the assessment result by the experts for the system modelling practices, we

conclude that the significant practices to be conducted are the item 3 (model the system architecture), item 1 (develop complementary system models), item 4 (use structured methods for system modelling), item 2 (model the system's environment) and item 6 (document the links between stakeholder requirements and system models).

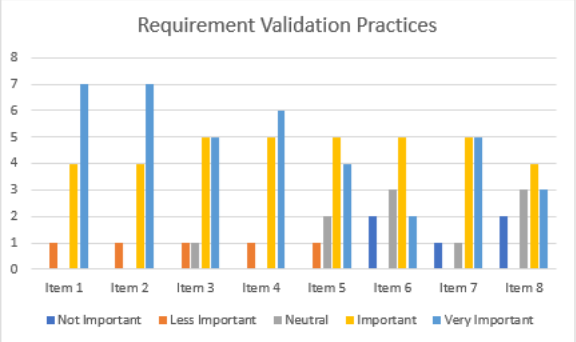


Item	Description
Item 1	Develop complementary system models
Item 2	Model the system's environment
Item 3	Model the system architecture
Item 4	Use structured methods for system modelling
Item 5	Use a data dictionary
Item 6	Document the links between stakeholder requirements and system models

Figure 9: Assessment result for System modeling practices

Requirement Validation Practices

Figure 10 shows the assessment result for requirement validation practices. There are eight practices have been assessed. Half or more of the respondents thought that item 1, item 2 and item 4 are very important practices to be conducted during the requirement validation phase. Five respondents also agreed that item 7 is also very important to be carried in requirement validation process. Many respondents also thought that item 3, item 4, item 5, item 6 and item 7 are important practices. However, there were 2 respondents decided that item 6 and item 8 are not important. Another 1 respondent thought that item7 is also not important. Based on the assessment result presented in Fig. 10, we conclude that the most important practices in requirement validation are the item 1 (check that the requirements document meets your standards), item 2 (organise formal requirements inspections) and item 4 (define validation checklists).

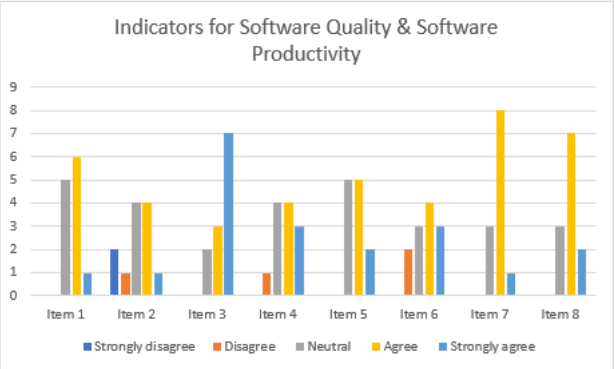


Item	Description
Item 1	Check that the requirements document meets your standards
Item 2	Organise formal requirements inspections
Item 3	Use multi-disciplinary teams to review requirements
Item 4	Define validation checklists
Item 5	Use prototyping to animate requirements
Item 6	Write a draft user manual
Item 7	Propose requirements test cases
Item 8	Paraphrase system models

Figure 10: Assessment result for Requirement validation practices

Indicators for Software Quality and Software Productivity

Respondents were also asked to assess the indicators for software quality and software productivity. There were eight indicators need to be assessed by the respondents. Figure 11 depicted the assessment result of the indicators for software quality and software productivity. More than half of the respondents (7 respondents) voted Strongly Agree for item 3 as the most significant indicator for software quality and software productivity. Half or more respondents voted Agree for item 7 (8 respondents), item 8 (7 respondents) and item 1 (6 respondents). There were 3 respondents disagree with item 2, 1 respondent disagree with item 4 and another 2 respondents disagree with item 6. Thus, based on the assessment result for the indicators for software quality and software productivity, we conclude that the most important indicators are the item 3 (end-users found the finished product was easy to use), item 7 (the ability and previous experience of the development team was adequate), item 8 (the quality of the development team's work was acceptable) and item 1 (project costs were within budget estimates).



Item	Description
Item 1	Project costs were within budget estimates
Item 2	Project goals were achieved earlier than predicted
Item 3	End-users found the finished product was easy to use
Item 4	The duration of the project was within schedule
Item 5	The team size was adequate for the project
Item 6	Error severity was not significant in the project
Item 7	The ability and previous experience of the development team was adequate
Item 8	The quality of the development team's work was acceptable

Figure 11: Assessment result for Indicators for software quality and software productivity

Summary of Results

This study was motivated from the legacy issues in Requirement Engineering such as poor in both requirement elicitation and requirement management. Thus, this empirical study intended to investigate and to discover the requirements engineering practices that the industry practitioners thought important and should be given high priority to ensure the software requirements obtained during RE are complete and correct. Table xx presented the summary assessment results of the study. We combined scale "Strongly Important" and "Important" as Significant, whereas the scale "Less Important" and "Not Important" as Insignificant. We presume that the items with 75% and above votes are the most important or the unimportant RE practices.

Table 1

The summary of assessment results

Category	Item #	Item Description	Insignificant (%)	Neutral (%)	Significant (%)
Elicitation	Item 1	Assess System Feasibility	0.0	16.7	83.3
	Item 2	Be sensitive to organizational and political consideration	25.0	16.7	58.3
	Item 3	Identify and consult system stakeholders	0.0	25.0	75.0
	Item 4	Record requirements sources	0.0	8.7	91.3
	Item 5	Define the system's operating environment	0.0	33.3	66.7
	Item 6	Use business concerns to drive requirements elicitation	8.3	16.7	75.0

	Item 7	Look for domain constraints	8.3	33.3	58.3
	Item 8	Record requirements rationale	0.0	25.0	75.0
	Item 9	Collect requirements from multiple viewpoints	8.3	25.0	66.7
	Item 10	Prototype poorly understood requirements	25.0	25.0	50.0
	Item 11	Use scenarios to elicit requirements	0.0	16.7	83.3
	Item 12	Define operational processes	8.3	16.7	75.0
	Item 13	Reuse requirements	25.0	25.0	50.0
Analysis and negotiation	Item 1	Define system boundaries	0.0	16.7	83.3
	Item 2	Use checklists for requirements analysis	16.7	25.0	58.3
	Item 3	Provide software to support negotiations	16.7	25.0	58.3
	Item 4	Plan for conflicts and conflict resolution	8.3	16.7	75.0
	Item 5	Prioritise requirements	0.0	16.7	83.3
	Item 6	Classify requirements using a multidimensional	8.3	16.7	75.0
	Item 7	Use interaction matrices to find conflicts and overlaps	8.3	33.3	58.3
	Item 8	Assess requirements risks	8.3	16.7	75.0
Describing Requirement	Item 1	Define standard templates for describing requirements	8.3	0.0	91.7
	Item 2	Use languages simply and concisely	8.3	0.0	91.7
	Item 3	Use diagrams appropriately	8.3	0.0	91.7
	Item 4	Supplement natural language with other description of requirement	8.3	8.3	83.3
	Item 5	Specify requirements quantitatively	8.3	16.7	75.0
System Modelling	Item 1	Develop complementary system models	8.3	25.0	66.7
	Item 2	Model the system's environment	8.3	8.3	83.3

	Item 3	Model the system architecture	8.3	8.3	83.3
	Item 4	Use structured methods for system modelling	8.3	8.3	83.3
	Item 5	Use a data dictionary	8.3	16.7	75.0
	Item 6	Document the links between stakeholder requirements and system models	8.3	0.0	91.7
Validation	Item 1	Check that the requirements document meets your standards	8.3	0.0	91.7
	Item 2	Organise formal requirements inspections	8.3	0.0	91.7
	Item 3	Use multi-disciplinary teams to review requirements	8.3	8.3	83.3
	Item 4	Define validation checklists	8.3	0.0	91.7
	Item 5	Use prototyping to animate requirements	8.3	16.7	75.0
	Item 6	Write a draft user manual	16.7	25.0	58.3
	Item 7	Propose requirements test cases	8.3	8.3	83.3
	Item 8	Paraphrase system models	16.7	25.0	58.3
Indicators for Software Quality & Software Productivity	Item 1	Project costs were within budget estimates	0.0	41.7	58.3
	Item 2	Project goals were achieved earlier than predicted	25.0	33.3	41.7
	Item 3	End-users found the finished product was easy to use	0.0	16.7	83.3
	Item 4	The duration of the project was within schedule	8.3	33.3	58.3
	Item 5	The team size was adequate for the project	0.0	41.7	58.3
	Item 6	Error severity was not significant in the project	16.7	25.0	58.3
	Item 7	The ability and previous experience of the development team was adequate	0.0	25.0	75.0
	Item 8	The quality of the development team's work was acceptable	0.0	25.0	75.0

1. Requirements Elicitation: Our findings show that out of 13 practices, 7 practices voted 75% and above for Significant. The highest vote is Item 4 (record requirement sources). Almost all experts in this study agreed that the sources of the requirements are very important and must be recorded. This is to ensure the requirements are legit and reliable. The next two items voted as the second highest are Item 1 (Assess System Feasibility) and Item 11 (Use scenarios to elicit requirements). This result shows that experts agreed to assess the feasibility of the system before the system can actually been developed. This is to avoid the waste of resources such as money, time and effort for developing an impossible or useless system. The experts also thought that the use of scenarios to obtain requirements is also important because through scenario, the system analyst can know and understand the flow of a certain business process and also person involve in that process. The other practices voted as significant are the Item 3 (Identify and consult system stakeholders), Item 6 (Use business concerns to drive requirements elicitation) and Item 12 (Define operational processes).

2. requirements analysis and negotiation: We found that out of 8 practices, 5 practices obtained 75% and above votes for Significant. Item 1 (Define system boundaries) and Item 5 (Prioritise requirements) voted as the most significant practices. The experts agreed that it is crucial to define the system boundary. This is to ensure the system is developed within the mutual agreement in between the developer and the client. By knowing this, the system analysts and the requirement engineers know to what extend they have to go to obtain the requirements. And then, they must prioritise the requirements. The other 3 practices voted as significant are the Item 4 (Plan for conflicts and conflict resolution), Item 6 (Classify requirements using a multidimensional approach) and Item 8 (Assess requirements risks).

3. Describing Requirement: The assessment results shows that all 5 practices in this category are voted with 75% and above as Significant. Thus, the experts agreed that to properly describe the requirements, we must have the standard template, use simple and concise language, use diagram appropriately, provide appropriate description for the requirements and specify requirements quantitatively.

4. System Modelling: As for System Modelling, 5 out of 6 practices obtained 75% and above votes as Significant. Item 6 (Document the links between stakeholder requirements and system models) voted as the highest and for that, we presume this practice is the most important for System Modelling. The experts thought that the linkage between requirements and system models must be documented. This is due to keep track the where about of each requirement in the system. Other than that, we do not want to miss any obtained requirements. The second highest votes are the Item 2 (Model the system's environment), Item 3 (Model the system architecture) and Item 4 (Use structured methods for system modelling). The experts also agreed that it is important to model the system's environment and system's architecture and this system modelling must be done using structured method. By doing so, we could foresee the to-be-developed system for any potential risks of faults or unsuccessful system.

5. Requirement validation: Our finding shows that 6 practices out of 8 got 75% and above votes as Significant. Item 1 (Check that the requirements document meets your standards), item 2 (Organise formal requirements inspections) and item 4 (Define validation checklists) obtained 91.7% votes. The experts thought that the requirements document must properly

presented at certain standard and the requirements must go through formal inspections. The validation checklist must also be defined. These practices are important to ensure the process of validating the requirements runs smoothly and the chances of overlooking at any angle of the requirements can be reduced or eliminated. Item 3 (Use multi-disciplinary teams to review requirements) and item 7 (Propose requirements test cases) are the second highest voted by the experts. The result shows that it is important to have people from multiple area concerning the requirements to be validated to involve in the validation process. This is very important because we want to get feedback from all groups of people. As for item 7, the test cases are also important during requirements validation process to ensure the validation process covers all angles of the requirements. The other practice obtained 75% votes is item 5 (Use prototyping to animate requirements). By having a prototyping, the requirements could be understood even clearer.

6. Indicators for Software Quality & Software Productivity: This is not part of the RE practices as in the Sommerville's framework. However, we would like to know what indicators the experts perceive to tell them that the software is doing good and performing as expected by the users. The study found that only 3 items managed to obtain 75% and more votes as Significant, which are item 3 (End-users found the finished product was easy to use), item 7 (The ability and previous experience of the development team was adequate) and item 8 (The quality of the development team's work was acceptable). The highest vote is item 3. Obviously, a software is doing well when the end-users can use it easily. As for item 7 and item 8, the experts thought the development team's ability, experiences and work quality are also important to indicate the software quality and software productivity.

Conclusion and Future Work

To conclude this discussion, there is no single approach that is best suited the RE process in software development project. It is also indirectly shows that there is no specific approach could solve the problem in the traditional RE process. Therefore, the current crowd-based RE research span through approaches, techniques, tools, and web-based platforms able to assist requirements engineering process in many ways while utilizing crowdsourcing benefits. Each of the efforts is unique in a way to solve a specific problem or to address the explicit concern in any of the requirements engineering areas. The efforts are made to take the advantage of crowdsourcing technique and the assistance of the AI technique, to obtain quality requirements.

For future works, it is beneficial to explore the utilization of crowd-based RE to obtain quality software requirements by optimizing the depth and breadth of information at reduced cost of time and money. We believe crowd-based RE can simplify and improve RE process in order to obtain quality software requirements which later, able to produce quality software systems.

References

Abran, A., Bourque, P., Dupuis, R. and Moore, J. W. (2004). Swebok. Guide to the Software Engineering Body of Knowledge.

Emam, K. E., & Madhavji, N. (1995). A field study of requirements engineering practices in information systems development. Proceedings of the Second IEEE International Symposium on Requirements Engineering, 68-80.

Ghanyani, U. S., Murad, M. and Mahmood, W. (2018). Crowd-based requirement engineering. International Journal of Education and Management Engineering, 3, 43-53.

Hall, T., Beecham, S., & Rainer, A. (2002). Requirements problems in twelve software companies: an empirical analysis. IEEE Software Proceedings. 149(5), 153-160.

Hussain, A., Mkpojiogu, E. O., and Kamal, F. M. (2016). The role of requirements in the success or failure of software projects. International Review of Management and Marketing, 6(7), 306-311.

Khan, H. U., Niazi, M., El-Attar, M., Ikram, N., Khan, S. U. & Gill, A. Q. (2021). Empirical investigation of critical requirements engineering practices for global software development. IEEE Access, 9, 93593-93613.

Lafta, I. A. (2015). A Study of Requirements Engineering Practices Among Software Developers at UUM Information Technology (UUMIT) (Doctoral dissertation, Universiti Utara Malaysia).

Niazi, M., & Shastry, S. (2003). Role of requirements engineering in software development process: An empirical study. Proceedings of the 7th International Multi Topic Conference, 402-407.

Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., Dar, S.H. and Khalid, S. (2021). Requirement engineering challenges in agile software development. Mathematical Problems in Engineering, 2021(1), 6696695.

Sadraei, E., Aurum, A., Beydoun, G., & Paech, B. (2007). A field study of the requirements engineering practice in australian software industry. Requirements Engineering, 12(3), 145-162.

Solemon, B., Sahibuddin, S., & Ghani, A. A. A. (2008). Requirements engineering problems in 63 software companies in Malaysia. Information Technology, 1-6.

Sommerville, I., & Sawyer, P. (1997). Requirements Engineering: A Good Practice Guide, John Wiley & Sons.

Bakar, W. M. L. & Solemon, B. (2017). Exploring the public's perception towards crowdsourced crime reporting.