High Perceived-Value Requirements Engineering Practices for Outsourced Software Projects

Rodina Ahmad[1]*, Mohd Hairul Nizam Md Nasir[1], Javed Iqbal[1] , Salmi Md Zahid[2]

[1] Faculty of Computer Science & Information Technology, University Malaya, 50603 Kuala Lumpur, Malaysia.
[2] Faculty of Management, Multimedia University, Malaysia.

* Corresponding author. Tel.: +603 79676302; email: rodina@um.edu.my

Abstract: This paper identifies high perceived-value requirements engineering (RE) practices specifically for outsourced software development projects. This is part of a study that aims to develop a framework for critical requirements engineering (RE) practices for outsourced software development projects. We used a survey questionnaire as the primary approach to collect data from 193 software development practitioners from 62 companies in Malaysia. The practitioners were asked to identify which RE practices, out of the 66 practices of Sommerville's framework, are considered critical for outsourced projects. Specifically, the practitioners were asked to assess and categorize the practices into four categories: high, medium, low and zero value. Only two RE practices were assigned a 'high value' by the practitioners and another 26 practices were assigned a 'medium' value. The findings show that there is an increased awareness of the importance of RE practices among Malaysian practitioners because only a very low percentage of the respondents selected most of the practices as zero- or low-value practices. Secondly, all the 28 practices ranked as 'high' and 'medium' value practices are critical and should be given high emphasis in the outsourced projects.   Moreover, due to the many more specific characteristics of outsourced projects, the study also implies that there is an urgent need to identify other RE practices or other related software engineering practices that are critical for outsourced projects but that have not been captured within Sommerville's framework.

Key words: Empirical study, requirement engineering, software projects, software development outsourcing.

1. Introduction

Requirements engineering (RE) problems are universal and indeed very significant [1]. Numerous studies have demonstrated the importance of focusing on the RE process for successful software projects; RE also affects the quality of the software product. It has been noted that there is a clear relationship among requirements gathering, analysis, and software quality [2]. The importance of RE lies in the importance of the product itself. Improving the RE process has the potential to reduce development costs and time and thus increase the quality of the software system [3]. A study of requirements errors in safety-critical embedded systems demonstrated that 60% of the critical system errors were the result of requirements errors and failures [4]. Additionally, in an empirical study regarding RE, Hell et al. [5] found that out of a

total of 268 cited development problems, almost 50% (128) were requirements problems.

The widespread use of Internet technologies and communication media has opened many opportunities for more communication and collaboration among people anywhere in the world. This trend has led to more widespread outsourcing projects in which a company contracts out all or part of its software development activities to another company that provides services for remuneration. According to Boehm [6], the primary motivation behind outsourcing is to decrease the overall cost.

1) There are four different scenarios for IT outsourcing [7]:
2) When a contractor provides services at the location of the outsourcing organization.
3) In the case of onshore outsourcing or domestic outsourcing, services are not provided at the outsourcing location, but the contractor operates from the same country.
4) The contractor provides services from another country: if services are provided from the same region/nearby country, it is called near shoring, but if the vendor supplies services from a distant country, it is called offshoring.

When multiple contractors/vendors are involved, if the stakeholders are geographically dispersed, it is called distributed software development (DSD). When the stakeholders are distributed around the world, it is called global software development (GSD) [8].

In spite of all the technological advancements and the expenditure of essential resources, the failure rate of outsourced projects is high. High turnover of staff, resistance to change, hidden costs, expansion of scope, refutation of the contractor from the service provision, use of outdated technology by the vendor, legal issues, cultural differences, and high duties and tax rates at the vendor's location are some of the most common risks for outsourced projects. However, the risks related to software requirements are the primary reason for the failure of outsourced projects [9]. This is a poor state of affairs because requirements engineering (RE) is the most important phase of the software development life cycle [10], and it affects the other software development activities quite significantly [11].

Although there are many RE practices that have been identified as important to ensure a successful implementation of software development projects, there are very few studies that focuses on identifying critical RE practices for outsourced software development projects. We intend to adapt the 66 RE practices suggested in the Sommerville and Ransom framework for outsourced software projects. The objective of this paper is to report an empirical study which aims to identify high perceived value RE practices for outsourced projects. The following research question is addressed:

RQ: Which RE practices are suitable to be used in outsourced software projects?

Hence, to address this research question, we have to identify the most critical RE practices from the 66 practices suggested in Sommerville's framework which supports outsourced software projects. Moreover, we need to identify other additional RE practices which may not be available from the framework.

The remainder of this paper is structured as follows: Section 2 presents the background on RE practices. Section 3 explains the design of our study, which includes the survey instrument used and the data collection approach. In Section 4, our findings are presented, analysed and discussed. Section 5 presents a summary of the findings for the study. Section 6 concludes the study with an outline for future work.

2. Related Work

There are a number of 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 (either development environments or methodology [12], [13], [14], [15].

One of the first works in this field was a study by El-Emam and Madhavji [3]. Their work explained the

issues that can effect 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. It was noted that organizational and project problems, or non-technical problems, have a direct impact on the requirements engineering process. Hofmann and Lehner's study of requirements engineering and project success also supported these findings and recommended that during the early stages, managers of successful RE teams should consult experts to augment and validate the team's knowledge base [16].

Hall *et al.* [5] discussed the results of a survey regarding RE problems in 12 different companies. The authors noted that the process of gathering and analyzing requirements is the key problem in the software development process. It was found that RE caused 48% of all discovered problems. Sadraei *et al.* [17] 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 completely separate process. It also found that the awareness of documenting requirements increased with project size (i.e., 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.

Niazi and Shastry identified the roles of RE in the software development process [2]. Their work was a detailed empirical study of requirements problems identified by 11 Australian software companies. The study was based on interview sessions with 22 practitioners from those companies. The findings were grouped into two categories: problems faced by organizations with a mature requirements 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 [2]. [18], 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.

Badariah et al. [19] 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.

Another study conducted by Boehm *et al.* [20] examined the requirements negotiation process for an internal digital library application project. The project involved over 50 professional librarians as clients and 5 or 6 computer science graduate students as software engineers. The authors elucidated that the 'two-culture' problem was the key challenge to determine a satisfactory set of requirements for the application under development. These 'two cultures' were the two groups of customers, which they named as the simplifiers and the complications. The authors argued that this approach reduced requirements problems within the project that they examined. This approach also reduced the number of projects that had serious problems regarding the feasibility of the Life Cycle Objectives milestone (for a series of projects between 1996 and 1998) from 4 out of 15 in 1996 and 4 out of 16 in 1997 to 1 out of 20 in 1998   [20].

Regarding RE practices, Cox et al. [21] empirically studied the use of Sommerville and Sawyer's RE practices in 10 Australian companies. The authors identified two sets of practices. The first set was used by organizations as a standard across all projects, and the other set was used only within one project. The

authors considered that organizations use RE practices because of their perceived value for that practice. However, some improvements in the practice guidelines were suggested by Sommerville and Sawyer [18].

3. Research Design

This study used a questionnaire survey method to gather data regarding critical RE practices for outsourced software development projects. This method is highly recommended for software engineering projects; especially to understand current practices or trends [22] The RE practices investigated were selected based on Sommerville and Sawyer's RE good practices guidelines [23]. In their work, the authors suggested a list of 66 RE practices that can help practitioners implement a better RE process. These RE practices were divided into three main groups: basic (36 practices), intermediate (21 practices), and advanced (9 practices). This categorization was based on the RE process maturity model [18].

3.1. Survey Instrument

In this study, each respondent was asked to assess and ranked 66 RE practices into four categories that have been developed previously [1]. These categories are the following:

1) High perceived benefit: A RE practice is referred to as having 'high perceived benefits' if it has a documented standard and is always followed as part of an organization's process to address outsourcing, i.e., it is mandatory.
2) Medium perceived benefit: A RE practice is referred to as having 'medium perceived benefits' if it is widely followed in the organization's process to address outsourcing but is not mandatory.
3) Low perceived benefit: A RE practice is referred to as having 'low perceived benefits' if it is introduced only for some projects.
4) Zero perceived benefit: A RE practice is referred to as having 'zero perceived benefits' if it is never or rarely applied to any outsourced project.

From this list, we have the 'perceived benefit' associated with each RE practice, i.e., the degree of importance placed on a RE practice by respondents based on their experience from previous software outsourcing projects. Other similar research has adapted the definition for 'perceived value'; for example, [24] used 'perceived value' to mean the extent to which a practice is perceived to provide value to a project or an organization based on the perceptions of practitioners whom are involved in the area of SPI. Here, we have provided an example of how we used an assessment scale, which is depicted in Table 1.

Table 1. Example of an Assessment Scale

RE Practices	H	M	L	Z
Lay out the document for readability. The layout of the requirements document should be designed so that it is easy to read and avoids messy representation of text.				

H=High Perceived Benefit; M=Medium Perceived Benefit; L=Low Perceived Benefit; Z= Zero Perceived Benefit

The data collection method significantly influences the data analysis process that can be used in the research [25]. 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 Niazi [24] used a similar approach. The questionnaire was developed based on Sommerville and Sawyer's RE good practices guidelines.

To uncover potential problems in the design and application of the questionnaire, we ran a pilot study that involved two researchers and two industry practitioners in the field of software development projects;

they were chosen using a convenient sampling method. Prior to releasing the questionnaire, [26] ran a test on each item to assess the respondents' level of understanding, level of knowledge, and level of difficulty in responding and the level of relevance of the item to the subject area. 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. This did not become a significant issue because we decided to use a face-to-face questionnaire-based survey session in which the respondents have the opportunity to ask questions if they do not understand something during the session. 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: Respondents were targeted specifically for their level of knowledge and experience in the area of software development outsourcing projects. Additionally, we included an empty space at the end of every page of the questionnaire for the respondents to identify and state any additional requirement management practices that they felt were relevant.

3) Level of difficulty: Our pilot study did not highlight any problems in responding to the questions. Thus, we did not face any problem 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.

3.2. Data Collection Implementation

This study primarily used face-to-face meetings to distribute the survey questionnaires and collect them. Although it was possible to distribute the questionnaires online, we chose to meet with the respondents face-to-face to enable us to provide further clarification and explanation to the respondents and increase the validity of the data. We distributed the questionnaire to approximately 300 respondents, from which 193 completed questionnaires managed to be collected, representing a response rate of 63.3% of respondents who have answered the questionnaires. Only 10 of the survey questionnaires were received through e-mail. These 193 respondents are software development practitioners from 62 companies in Malaysia.

4. Findings

The questionnaire was designed to gather data regarding the perceived value of practices from the eight main groups of practices based on Sommerville's framework, which are requirements documentation, requirements elicitation, requirements analysis and negotiation, describing requirements, system modeling, requirements validation, requirements management, and RE practices for critical systems. However, in this paper, we do not discuss the RE practices for critical systems as not all participants have experience developing such systems. For the analysis of the perceived value of each RE practice, the occurrence of a perceived value (high, medium, low, or zero) 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.

4.1. Respondent Demographics

Fig. 1 shows the distribution of our respondents according to their position in their company. We have grouped the respondents into 3 groups: 1) a senior executive/executive-level group that consists of requirements engineers, analysts, programmers, designers, and software engineers; 2) a middle management-level group, which consists of project managers, department heads, and technical consultants; and 3) a top management-level group, which consists of directors and general managers who are involved in decision-making. Most respondents hold positions at the senior executive/executive level; this group accounted for 78.8% (152/193) of the respondents. Approximately 19.7% (38/193) of the respondents hold middle management-level positions, and the remainder, 1.6%, holds top management-level positions. Although our survey includes a high fraction of senior executive/executive-level respondents, the respondents in this group are not junior executive or inexperienced: We confirmed that they have at least 5 years of experience in software outsourcing projects. Additionally, all of the respondents address real software outsourcing project issues on a daily basis; thus, the sample is not random but should rather be considered a convenience sample.
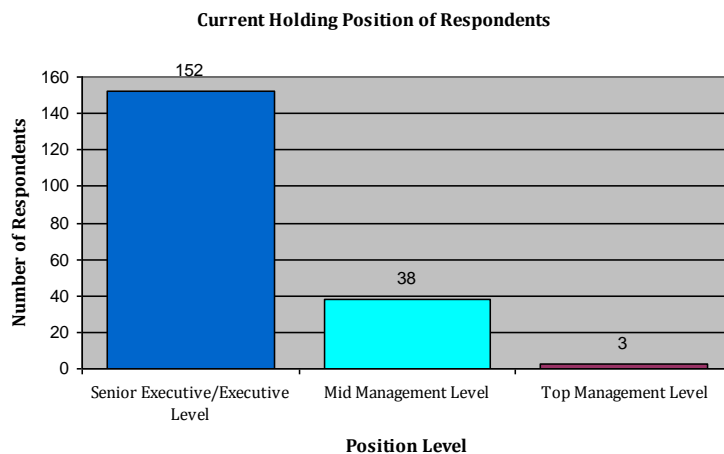
Current Holding Position of Respondents

Fig. 1. Current position of respondents.

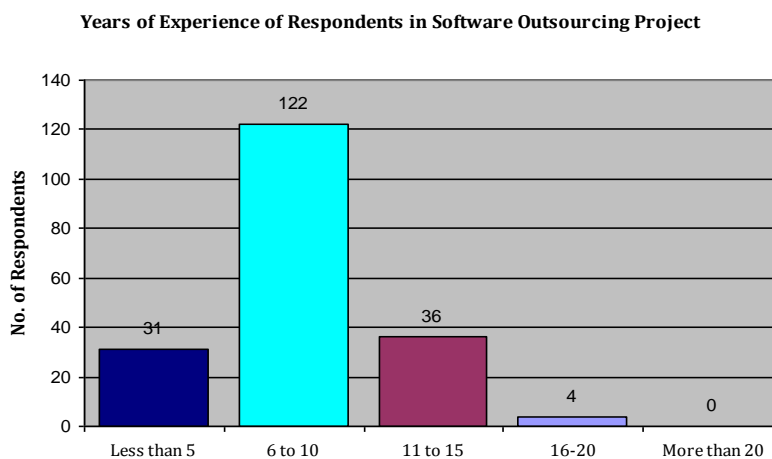**Years of Experience of Respondents in Software Outsourcing Project**

Fig. 2. Respondents' years of experience with software outsourcing projects.

The experience of these respondents in software outsourcing projects is illustrated in Fig. 2. Approximately 16.1% (31/193) of the experts have less than 5 years of experience, whereas 18.7% (36/193) have 11-15 years of experience. The majority of respondents, 63.2% (18/30), have 6-10 years of experience, which indicates a non-trivial background for performing this assessment. Finally, 2.1% (4/193) of the respondents have 16-20 years of experience.

The respondents were chosen from organizations of different sizes and with a wide range of products and markets that develop either software or combined software and hardware products. The organizations include telecommunication providers, security providers, system integrators, document solutions providers, finance and banking solutions' providers, among others. Based on the organization size definition provided by the National SME Development Council, Malaysia, for the ICT sector (micro: less than 5 employees, small: 5-19 employees, medium: 20-50 employees, and large: more than 50 employees), our respondents include 9 practitioners from micro-sized organizations, 55 practitioners from small-sized organizations and 92 practitioners from large organizations (as illustrated in Fig. 3).
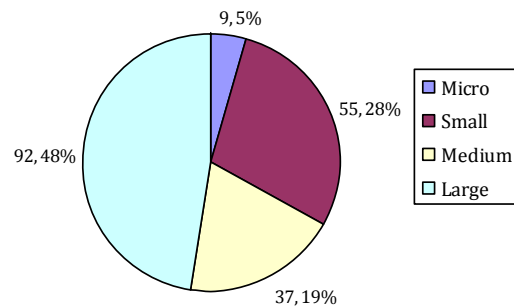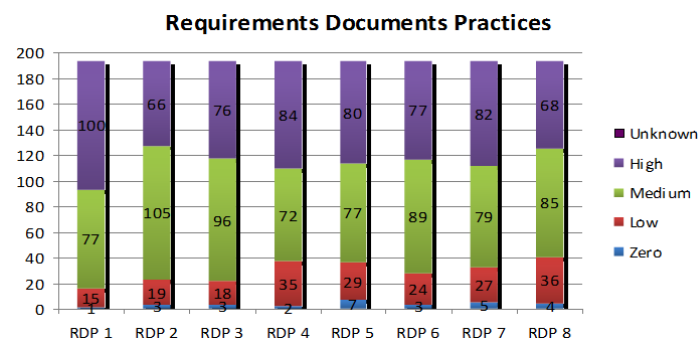

Fig. 3. Organization size of the respondents.

ID	Descriptions
RDP1	Define and follow a standard document structure
RDP2	Explain how to use the document
RDP3	Include a summary of the requirements
RDP4	Make a business case for the system
RDP5	Define specialized terms
RDP6	Make document layout readable
RDP7	Help readers find information
RDP8	Make the document easy to change

Fig. 4. Assessment results for requirements documents practices category.

As mentioned previously, these RE activities contain a set of best practices that were described in [18]. The following section discusses the RE practices covered in this survey.

4.2. Requirements Documentation Practices

Requirements documentation practices include the primary practices that are considered to be important for managing changes to a system's requirements [27]. Requirements must evolve because of changes to a

system's environment or as customers develop a better understanding of their needs. Hence, there is a significant need to ensure that changes to the requirements are properly updated and documented. Fig. 4 presents the findings of the 8 practices classified into this category that were assessed and ranked by the Malaysian practitioners.

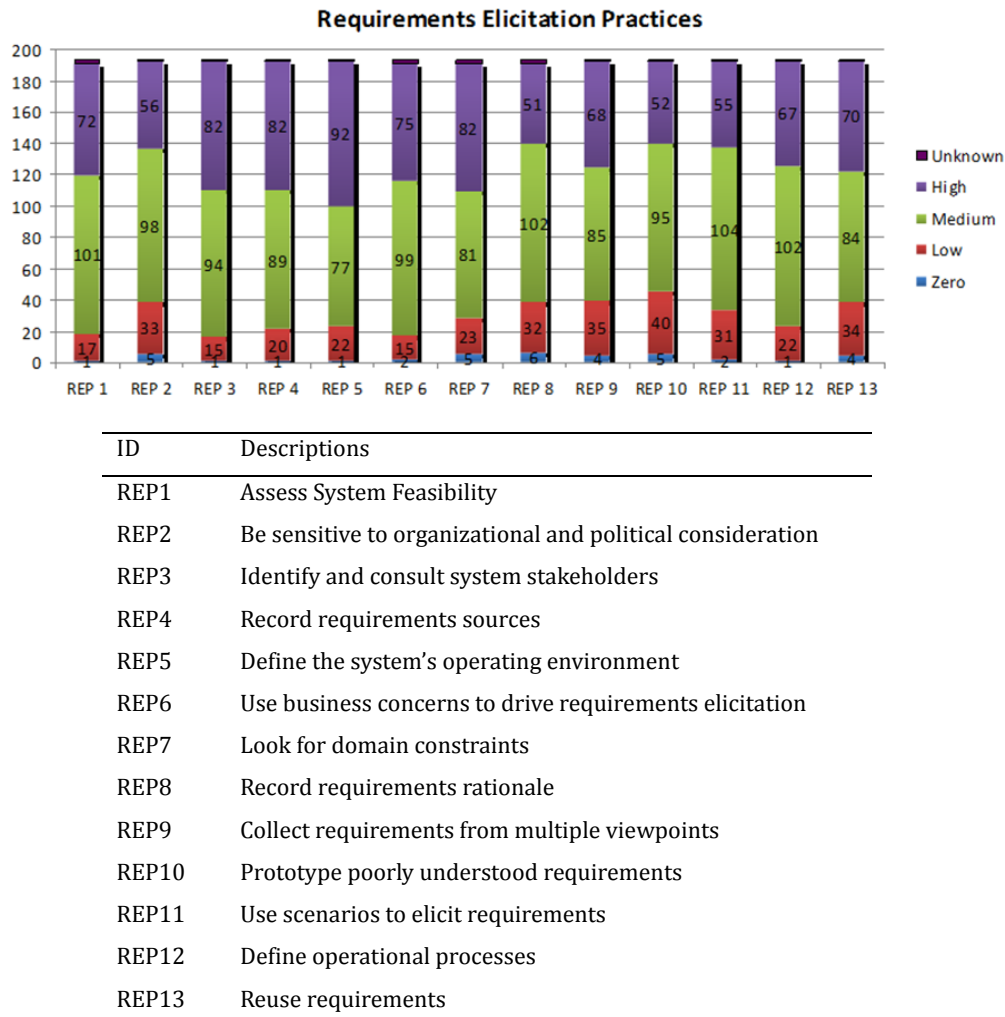| ID	Descriptions
REP1	Assess System Feasibility
REP2	Be sensitive to organizational and political consideration
REP3	Identify and consult system stakeholders
REP4	Record requirements sources
REP5	Define the system's operating environment
REP6	Use business concerns to drive requirements elicitation
REP7	Look for domain constraints
REP8	Record requirements rationale
REP9	Collect requirements from multiple viewpoints
REP10	Prototype poorly understood requirements
REP11	Use scenarios to elicit requirements
REP12	Define operational processes
REP13	Reuse requirements

Fig. 5. Assessment results for requirements elicitation practices category.

The results demonstrate that only one practice ('define and follow a standard document structure') was classified as 'high perceived value' by more than 50% of the respondents. The other 7 practices were classified as 'high perceived value' by between 34% and 44% of the respondents. Moreover, if we analyze all 8 practices, we notice that the majority of the respondents (more than 80% of the respondents) rank all 8 practices to be either 'medium perceived' or 'high perceived' value practices. These findings seem to indicate that the 8 practices that we surveyed were perceived to be important to the practitioners, especially in the context of outsourced projects.

4.3. Requirements Elicitation Practices

Requirements elicitation refers to the activities involved in searching, gathering and discovering the requirements of a system. It involves more than just asking people; it also requires a thorough analysis of the organization, the application domain and the business process [27]. Hence, this activity is very crucial, especially for outsourced projects in which the stakeholders may come from different organizations with diverse cultural backgrounds. Fig. 5 presents the findings for the 13 practices within this category.

Requirements Analysis and Negotiation Practices

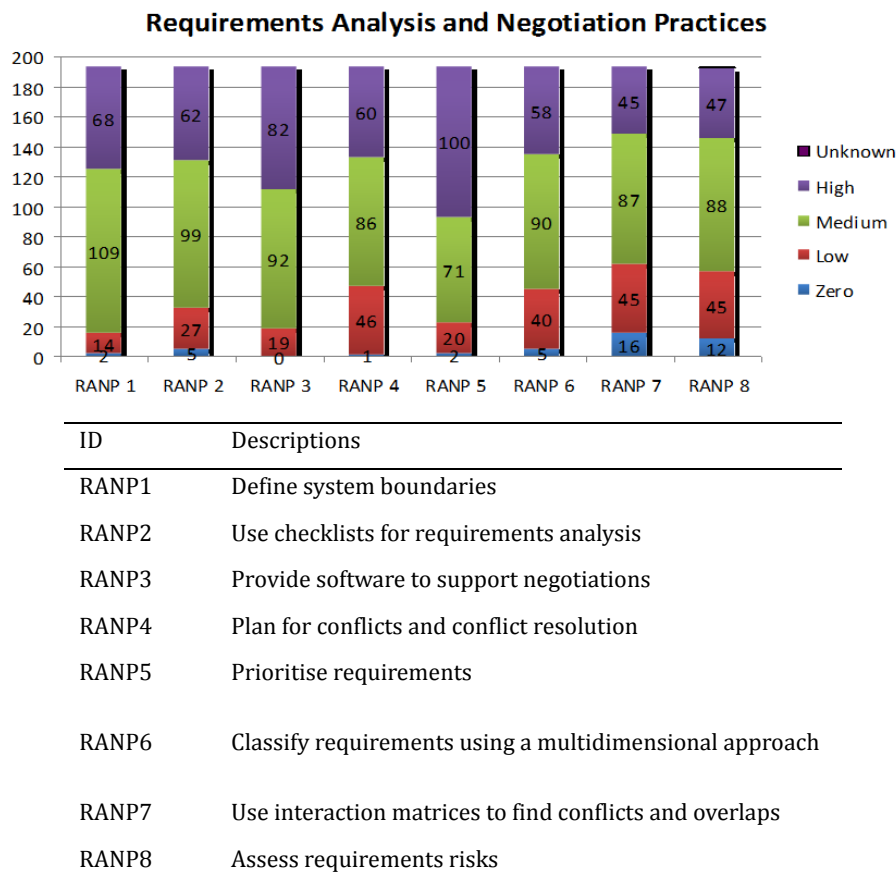| ID	Descriptions
RANP1	Define system boundaries
RANP2	Use checklists for requirements analysis
RANP3	Provide software to support negotiations
RANP4	Plan for conflicts and conflict resolution
RANP5	Prioritise requirements
RANP6	Classify requirements using a multidimensional approach
RANP7	Use interaction matrices to find conflicts and overlaps
RANP8	Assess requirements risks

Fig. 6. Assessment results for requirements analysis and negotiation practices category.

The results demonstrate that there is not one practice that was identified as a 'high perceived value' practice by more than 50% of the respondents. Instead, we found that for each practice of the requirements elicitation practices category, the majority of the respondents (greater than 80%) identified each of the practices as either a 'high perceived value' or 'medium perceived value' practice. It is also good to note that 6 of the 13 practices within this category were identified as 'medium perceived value' practices by more than 50% of the respondents. For all 13 practices within this category, the majority of the respondents (between 70% and 80%), ranked these practices as 'medium' or 'high' perceived value. Only approximately 10%-20% ranked these practices as 'low' or 'zero' value.

The literature seems to emphasize the importance of the elicitation process, especially when the projects involve many participating bodies. For example, [28] reports that elicitation and communication of software requirements is one of the four challenges of globally distributed software development projects. Hence, these findings motivate us to identify other additional elicitation practices that are vital, especially in the context of outsourced projects.

4.4. Requirements Analysis and Negotiation Practices

The objective of requirements analysis and negotiation is to identify problems with the system requirements and to agree on changes to satisfy the stakeholders. The findings of the survey demonstrate that only one practice ('prioritize requirements') was ranked by more than 50% of respondents as a 'high perceived value' practice (as illustrated in Fig. 6).

Requirements prioritization is a decision-making activity that helps order requirements based on their importance to the customer [29]. Prioritizing requirements leads to finding and identifying defects, such as

underestimating, incorrectness, and ambiguousness in requirements [30], because these requirements are analyzed from different perspectives when reviewing requirements. Prioritizing requirements also assists project managers in resolving conflicts, planning for stage deliveries, and making essential trade-off decisions [31].

Overall, only 1% to 6% of the respondents described these requirements analysis and negotiation practices as low value. Hence, this finding reflects the awareness of the practitioners on the importance of these practices for software development projects. In the context of outsourced projects, requirements prioritization is more challenging and demanding because of the possible differences in views caused by the various stakeholders' backgrounds and possibly different cultures. Hence, ensuring the proper implementation of requirements prioritization within outsourced software projects can be considered one of the key measures to ensure that the requirements are understood and can be the primary determinant of project success.

4.5. Describing Requirements Practices

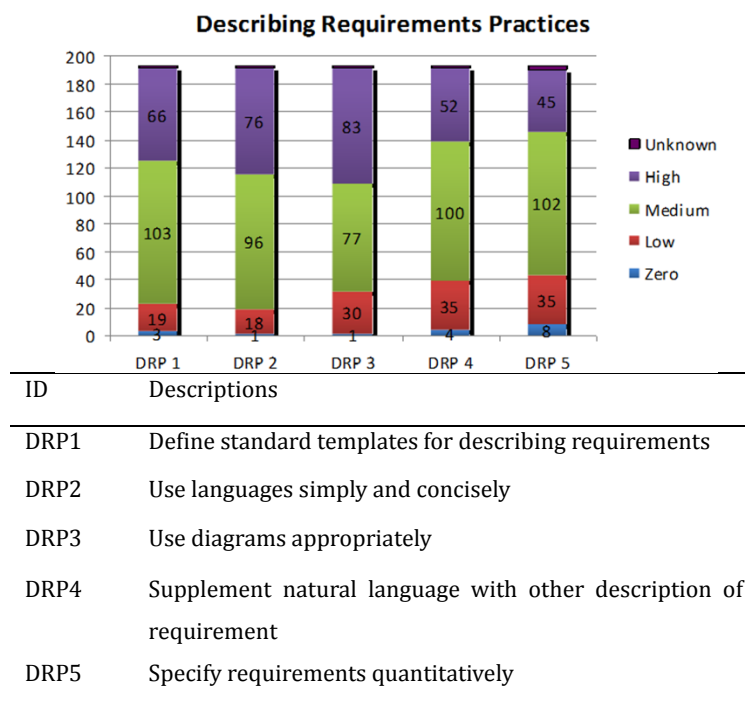| ID	Descriptions
DRP1	Define standard templates for describing requirements
DRP2	Use languages simply and concisely
DRP3	Use diagrams appropriately
DRP4	Supplement natural language with other description of requirement
DRP5	Specify requirements quantitatively

Fig. 7. Assessment results for describing requirements practices category.

The results demonstrate that there is not one practice under this category that was ranked by more than 50% of the respondents as a 'high perceived value' practice (see Fig. 7). However, 4 out of 5 of the describing requirements practices were identified by more than 50% of the respondents as 'medium perceived value' practices. It seems that in the context of outsourced projects, these practices may still be perceived as key practices that can help increase the efficiency or effectiveness of the RE process due to the differing background of stakeholders and users. Hence, having a proper and standard way of describing requirements will tremendously facilitate the process of achieving a shared understanding of the requirements from the variety of stakeholders.

4.6. System Modelling Practices

The aim of system modeling is to clarify, represent and check the validity of the requirements. The results indicate that there is not one practice in this category that was identified as 'high perceived value' by more than 50% of the respondents (see Fig. 8). However, 4 out of 6 of the system modeling practices were ranked

as 'medium perceived value' by more than 50% of the respondents. This finding indicates that although system modeling is critical in clarifying vague requirements, the practices may not be perceived as the primary practices to ease the process of requirements engineering in the context of outsourced software development projects, which might involve more differing, contrasting and diverse viewpoints.
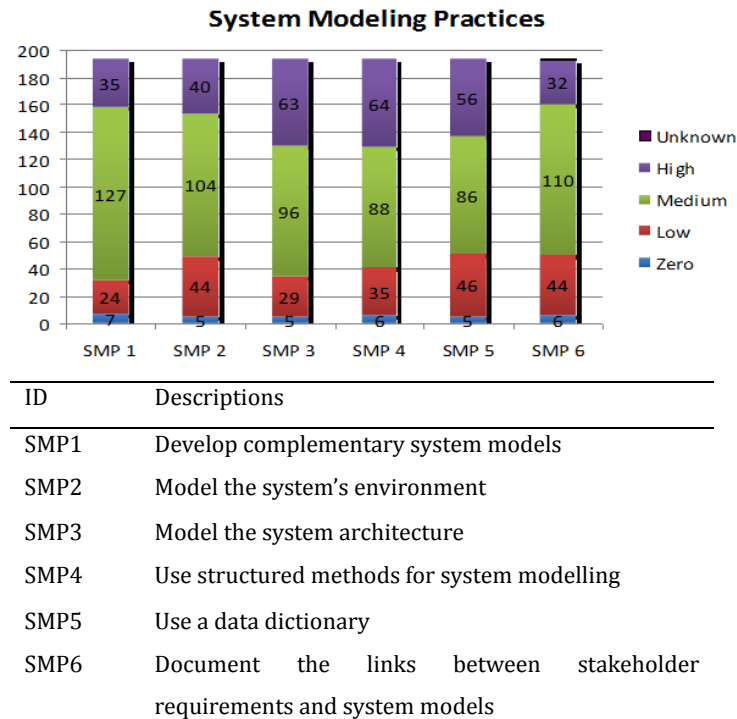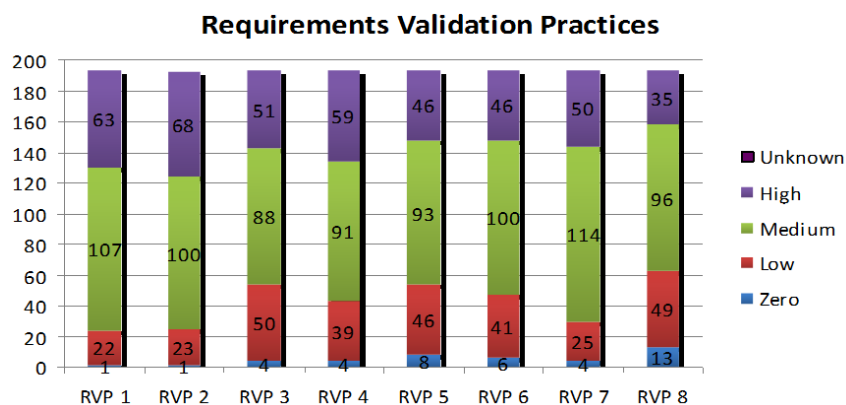

ID	Descriptions
SMP1	Develop complementary system models
SMP2	Model the system's environment
SMP3	Model the system architecture
SMP4	Use structured methods for system modelling
SMP5	Use a data dictionary
SMP6	Document the links between stakeholder requirements and system models

Fig. 8. Assessment results for system modelling practices category.

4.7. Requirements Validation Practices

The goal of requirements validation is to certify the requirements in order to ensure that they represent an acceptable description of the system to be implemented. The results for this category are quite similar to those for the previous two categories of practices, i.e. describing requirements and system modeling. No single practice of the requirements validation practices was ranked as 'high perceived value' by more than 50% of the respondents (see Fig. 9). However, 5 out of 8 practices were ranked as 'medium perceived value' by more than 50% of the respondents. Overall, 80% of the respondents ranked each practice as either a 'high perceived value' or 'medium perceived value' practice. These results seem to indicate the general importance of these practices.

ID	Descriptions
RVP1	Check that the requirements document meets your standards
RVP2	Organise formal requirements inspections
RVP3	Use multi-disciplinary teams to review requirements
RVP4	Define validation checklists
RVP5	Use prototyping to animate requirements
RVP6	Write a draft user manual
RVP7	Propose requirements test cases
RVP8	Paraphrase system models

Fig. 9. Assessment results for requirements validation practices category.

Fig. 10. Assessment results for requirements management practices category.
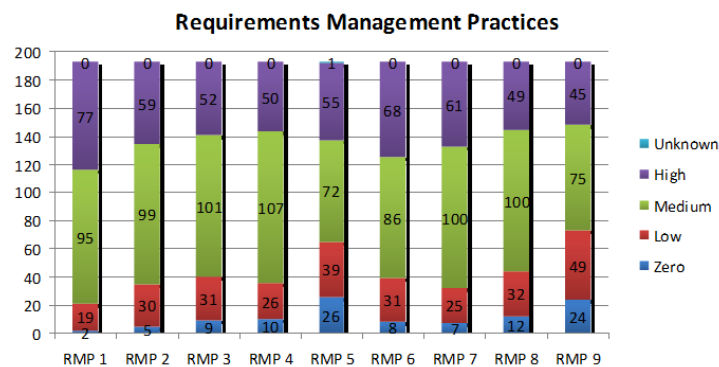
ID	Descriptions
RMP1	Uniquely identify each requirement
RMP2	Define policies for requirements management
RMP3	Define traceability policies
RMP4	Maintain a traceability manual
RMP5	Use a database to manage requirements
RMP6	Define change management policies
RMP7	Identify global system requirements
RMP8	Identify volatile requirements
RMP9	Record rejected requirements

4.8. Requirements Management Practices

The results indicate that not a single practice in this category was ranked as 'high perceived value' by more than 50% of respondents (see Fig. 10).

However, 5 out of 9 of these requirements management practices were identified as 'medium perceived value' by more than 50% of the respondents. It is interesting to note that for most practices within this category, only approximately 25% of the respondents ranked them as 'high perceived value'. The reason for this result might be the nature of the surveyed practices, which are intended to facilitate the process of managing individual specified requirements and do not actually address the possible complexity of managing requirements as a whole. However, these management practices can be considered to be more crucial that needs to be tackled appropriately especially for outsourced software projects because of the diversity of stakeholders, cultures and resources involved in such projects.

Table 2. High and Medium Value Practices

ID	Practice	Category	Rank
RD1	Define and follow a standard document structure	Documentation	High
RD2	Include a section in the introduction part of the document to explain how to use it		Medium
RE9	Assess system feasibility	Elicitation	Medium
RE10	Be sensitive to organizational and political considerations		Medium
RE14	Use business concerns to drive requirements elicitation		Medium
RE16	Records requirements rationale		Medium
RE19	Use scenarios to elicit requirements		Medium
RE20	Define operational processes		Medium
RAN22	Define system boundaries	Analysis and Negotiation	Medium
RAN23	Use checklists for requirements analysis		Medium
RAN26	Prioritize requirements		High
RP30	Define and use standard templates for requirements description	Description	Medium
RP31	Use simple, consistent and concise language to describe requirements.		Medium
RP33	Supplement natural language with other descriptions of requirements where appropriate.		Medium
RP34	Specify requirements quantitatively where appropriate.		Medium
SM35	1) Develop complementary system models	Modelling	Medium
SM36	Model the system's environment		Medium
SM40	Document the association between stakeholder requirements and system models.		Medium
RV41	Check that the requirements document meets your standard	Validation	Medium
RV42	Organize requirements inspection		Medium
RV46	Write a user manual draft		Medium
RV47	Propose requirements test cases		Medium
RM50	Define policies for requirements management	Management	Medium
RM51	Define requirements traceability policies		Medium
RM52	Maintain traceability manual		Medium
RM55	Identify global system requirements		Medium
RM56	Identify volatile requirements		Medium
RC63	Specify systems using formal specification.	Critical system	Medium

5. Summary of Results

This study was motivated by the challenges faced by outsourced software projects. Almost 40% of outsourced projects do not achieve the expected benefits, while approximately 50% of companies that have global software teams fail to obtain the anticipated results. Hence, this study intended to identify the critical requirements engineering practices that should be given high priority in order to ensure greater success for

outsourced software development. Although there are numerous practices mentioned in the literature that are suitable for outsourced projects, we decided to first ascertain the perception of practitioners towards the importance of the RE practices that were identified in Sommerville's framework. We created a survey instrument based on the RE practices listed in Sommerville's framework to gauge the importance of these RE practices for outsourced software projects.

In this study, we collected a set of data from software practitioners in Malaysia Based on the data collected,   only two RE practices were perceived to be of 'high' value, which means that the majority of the respondents (≥50%) ranked them as 'high value'. The results also list 26 practices as 'medium' value practices; these practices account for 40% of the total practices. Table 2 lists the practices for the 'high' and 'medium' value practices within each category.

6. Conclusion and Future Work

Beneficial requirements engineering practices are known to have a positive impact on software development projects. This empirical study intended to investigate software practitioners' perceptions regarding the importance of a selected set of RE practices for outsourced software development projects. The primary aim was to identify the critical RE practices to be recommended in order to help practitioners address the challenges of outsourced software projects. Essentially, this initial investigation has enabled us to obtain an understanding on the perception of software practitioners regarding the 66 RE practices based on Sommerville's framework. Our study considers that the 66 RE practices provided by the framework can be considered as the basic RE practices that are essential for diverse types of software development projects, including outsourced software development projects. However, because of the differing and challenging nature of outsourced projects, which have special characteristics, we have to identify more additional RE practices that can help address those challenges.

Our findings show that only two practices were ranked as high value practices and 26 practices were ranked as medium value practices. We presumed that the findings indicate that the participating respondents have already been exposed to a variety of challenges in outsourced projects and wish to see more specific RE practices that can cater to the specific challenges of outsourced software projects. Hence, this might be the reason why they ranked very few RE practices that we surveyed as the practices that can be potentially adapted to cater to outsourced projects.

In essence, this study enables us to recognize that there can be at least two categories of RE practices that can cater to outsourced projects. One set can come from the basic RE practices which have essentially been identified in Sommerville's framework. Another set of RE practices should try to focus on the specific challenges faced by outsourced projects. The following research questions are for our future work in this project:

RQ1: What other RE practices should be included in the Outsourced Requirements framework?

RQ2: How can we evaluate the Outsourced Requirements framework?

Our future work will attempt to identify the specific challenges and issues that outsourced software development projects face and the potential RE and general software engineering practices that can address these challenges. These practices must be gathered and validated using feedback from the software development experts as well as from the software practitioners to ensure that we are able to assist software developers in outsourced software development projects.

Acknowledgment

References

[1] Sommerville, I., & Ransom, J. (2005). An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology*, *14*, 85-117.

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

[3] 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* (pp. 68-80).

[4] Lutz, R. R. (1993). Analyzing software requirements errors in safety-critical, embedded systems in requirements engineering. *Proceedings of IEEE International Symposium* (pp. 126-133).

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

[6] Boehm, B. (2006). A view of 20th and 21st century software engineering. *Proceedings of the 28th International Conference on Software Engineering* (pp. 12-29).

[7] Gibbs, R. D. (2007). Project management with the IBM rational unified process: Lessons from the trenches.

[8] Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Computing Surveys*, *29*, 315-321.

[9] Oza, N. V., & Hall, T. (2005). Difficulties in managing offshore outsourcing relationships: An empirical analysis of 18 high maturity indian software companies. *Journal of Information Technology Case and Application Research*, *7(3)*, 25-41.

[10] Bhat, J. M., Gupta, M., & Murthy, S. N. (2006). Overcoming requirements engineering challenges: lessons from offshore outsourcing. *IEEE Software*, *23(5)*, 38-44.

[11] Ian, S., & Ransom, J. (2005). An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *14(1)*, 85-117.

[12] Maiden, N. A. M., & Sutcliffe, A. G. (1993). Requirements engineering by example: an empirical study," in Requirements Engineering. *Proceedings of IEEE International Symposium* (pp. 104-111).

[13] Kamsties, E., Hörmann, K., & Schlich, M. (1998). Requirements engineering in small and medium enterprises. *Requirements Engineering*, *3(2)*, 84-90.

[14] Karlsson, L., Dahlstedt, A. G., Regnell, B., Dag, J. N., & Persson, A. (2007). Requirements engineering challenges in market-driven software development — An interview study with practitioners. *Information and Software Technology*, *49(6)*, 588-604.

[15] Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: an empirical study. *IEEE Software*, *25(1)*, 60-67.

[16] Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, *18(4)*, 58-66.

[17] 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.

[18] Sawyer, P., Sommerville, I., & Viller, S. (1997). Requirements process improvement through the phased introduction of good practice. *Software Process Improvement and Practice*, *3(1)*, 19-34.

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

[20] Boehm, B., Antoun, M. A., Port, D., Kwan, J., & Lynch, A. (1999). Requirements engineering, expectations

management, and the two cultures. 1999, *Proceedings IEEE International Symposium on Requirements Engineering* (pp. 14-22).

[21] Cox, K., Niazi, M., & Verner, J. (2008). Empirical study of sommerville and sawyer's requirements engineering practices. *IET Software, 3(5)*, 339-355.

[22] Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287.

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

[24] Niazi, N., Cox, K., & Verner, J. (2006). An empirical study identifying high perceived value requirements engineering practices. *Advances in Information Systems Development*, 731-743.

[25] N. Baddoo and T. Hall, "De-motivators for software process improvement: an analysis of practitioners' views," *Journal of Systems and Software*, vol. 66, no. 1, pp. 23-33, 2003. doi:10.1016/S0164-1212(02)00060-2

[26] Berry, M., & Jeffery, R. (2000). An instrument for assessing software measurement programs. *Empirical Software Engineering*, *5(3)*, 183-200.

[27] Sommerville, I., & Kotonya, G. (1998). *Requirements Engineering: Processes and Techniques*, John Wiley & Sons, Inc.

[28] Mockus, A., & Herbsleb, J. (2001). Challenges of global software development. *Proceedings Seventh Symposium on Software Metrics* (pp. 182-184).

[29] Greer, D., & Ruhe, G. (2004). Software release planning: An evolutionary and iterative approach. *Information and Software Technology*, *46(4)*, 243-253.

[30] Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, *39(14)*, 939-947.

[31] Aurum, A., & Wohlin, C. (2005). Requirements engineering: Setting the context. *Engineering and Managing Software Requirements*.

Rodina Binti Ahmad is a lecturer and a researcher for more than twenty years. She received her bachelor degree from University of Hartford, Western Hardford, Connecticut in USA; the master degree from Rensselaer Polytechnic Institute in USA and completed her PhD in the area of information systems management from National University of Malaysia, Kuala Lumpur, Malaysia in 2006.

She is an associate professor at Software Engineering Department, University of Malaya, Kuala Lumpur, Malaysia. Her research interests include software process improvement, requirements engineering and organizational analysis (requirements elicitation, analysis, modeling, prioritization and evolution).

Dr. Rodina is a member of International Association of Computer Science and Information Technology and IEEE, and one of the PETRONAS scholarship holders.

Mohd Hairul Nizam Bin Md Nasir is from Kuala Lumpur Malaysia. He received his bachelor degree from University of Malaya, Kuala Lumpur, Malaysia, the master degree also from University of Malaya, Kuala Lumpur, Malaysia. He received his PhD from University Technology Malaysia in 2014.

He is a lecturer at Software Engineering Department, University of Malaya, Kuala Lumpur, Malaysia. His research interests include Software Process Models, Software Process Improvement (CMM, CMMI, TSP and PSP), Software Quality and Project Management (PMBOK, PRINCE2).

Mohd Hairul Nizam is member of Project Management Institute, Association for Computing Machinery (ACM) and IEEE.

Javed Iqbal is from Islamabad, Pakistan. Javed Iqbal completed his master in computer sciences from UAAR, Rawalpindi, Pakistan and Master in Software Engineering from National University of Science Technology, Islamabad, Pakistan. He is an assistant professor at Computer Science Department, COMSATS Institute of IT Islamabad, Pakistan. Currently he is pursuing his PhD at Software Engineering Department, University of Malaya, Kuala Lumpur, Malaysia. His areas of interest are software process improvement, requirement engineering and software development outsourcing. Javed has received Bright Sparks Program scholarship from University of Malaya, Kuala Lumpur, Malaysia.