International Journal of Software Engineering and Technology

Exploring Software Practitioners' Perceptions and Experience in Requirements Reuse

A Survey in Malaysia

Noor Hasrina Bakar and Zarinah M. Kasirun
Dept. of Software Engineering, FSKTM
University of Malaya, Kuala Lumpur, Malaysia

noorhasrina@siswa.um.edu.my, zarinahmk@um.edu.my

***Abstract*— In Software Product Lines (SPL) development, reuse process is planned ahead of time, while in traditional software development reuse can occur opportunistically: unplanned or in ad hoc manner. Although many research efforts in SPL focus on issues related to architecture, designs and codes reuse, research on requirements reuse has received slightly less attention from researchers and practitioners. Requirements Reuse (RR) in SPL is the process of systematically reusing previously defined and validated requirements for an earlier software product and applying them to a new and slightly different product within a similar domain. This paper presents a survey pertaining to RR practice that was conducted in Malaysia with two objectives: a) to identify the factors influencing software practitioners in RR, and b) to assess the factors hindering software practitioners from reusing requirements in software development. The survey results have confirmed seven factors that can influence RR practice in Malaysia. The survey results have also revealed three main impediments to RR practice in Malaysia: the unavailability of RR tools or framework to select requirements for reuse, the conditions of existing requirements to be reused (incomplete, poorly structured or not kept updated), and the lack of awareness and RR education among software practitioners pertaining to the systematic RR**

Keywords — Requirements reuse, software development, software product lines, systematic reuse.

I. Introduction

Reuse of software artefacts such as requirements, architecture, designs, codes, and test plans can produce many benefits including lowering development costs, increasing developers' productivity, and expediting time to market. This is true especially when reuse is considered early during software development. Requirement Reuse (RR) is the process of reusing previously defined and validated sets of requirements in a new development of software within a similar domain. Systematically, it is done in the context of Software Product Line (SPL) or Software Family. This is achievable by identifying commonality and variability requirements.

Various approaches were introduced to promote requirements reuse in SPL. For example, the work from feature and domain modelling appeared in FODA [1] and PLUSS [2]. Specifically, work on extracting requirements through features from natural language requirements appeared in [3][4]. Works in [5], [6] appeared on tackling requirements reuse through ontology approaches. Agent-based approach to RR appeared in [7]–[10] and a few more efforts on RR employed information retrieval techniques appeared in [11], [12].

Despite the known benefits, RR is not an easy task. Most requirements appeared in natural language [13] and they seem to be long in sentences [14]. Currently, tools that demonstrated the domain analysis which relates to RR from the SPL context were not widely available on the web, but rather published as research results from academia [15]. Some tools provided to aid RR activities were around DOORS extension [16]–[20]. In fact, very limited tools that dealt with reusing natural language requirements were not empirically validated [21]. Hence, research on how to handle natural language requirements in more automatic form is growing [22][23][24][25]. The field of Requirement Engineering in SPL has attracted many research initiatives for over two decades [21]. However, Alves et al. in [26] reported that the majority of the methods proposed resulted from research proposal rather than emerged from established industrial practice. RR was also reported to be in ad hoc manner [27] [28]. Even when reuse is planned, more work is needed to provide guidance to practitioners in the industry interested in adopting SPL [21].

In this study, we would like to investigate the current practice of RR in the Malaysian context. The objective of this study comes in twofold. Firstly, we are interested to identify the factors influencing individual software practitioners' perceptions and intentions to reuse software requirements in software development. Secondly, we want to assess the factors hindering software practitioners in Malaysia from reusing the existing requirements in new software development.

This paper is structured as follows: Section II discusses related works, Section III describes our research model and hypotheses, Section IV presents the survey constructions, Section V discusses the results, Section VI talks about threats to validity, and Section VII concludes this paper.

II. Related Works

In 1995, an empirical study was conducted to investigate the software practitioners' attitudes, beliefs, and practices in reusing codes and other software development artefacts [29]. In this study, Frakes et al. (1995) conducted a survey to answer sixteen common questions about software reuse in organizations within the US and Europe. The participants include software engineers, managers, educators, and other software development and research communities. Important findings from their survey revealed factors promoting systematic reuse include education about reuse, developers' understanding of the economic benefits of reuse, instituting common development process, and making high-quality assets available to developers.

Slyngstad et al. in [30] investigated the developers' view on software reuse through a survey conducted at Norway's Oil and Gas company. The study collected responses from 16 software developers at Statoil ASA. The results showed that reuse benefits from the developers' view include lower costs, shorter development time, higher quality of reusable artefacts, and a standardized architecture. Component understanding was found to be sufficient; however, an improvement to documentation is needed. In addition, they have found that there is no relation between reuse and increased rework.

Mellakord et al. in [31] conducted a study on multilevel analysis of factors affecting software developers' intention to reuse software assets in general. The survey was administered to 50 companies in India back in 2007. Technology Acceptance Model (TAM) from [32] was used in developing their conceptual research model. Results from [31] revealed that technological-level (infrastructure) and individual-level (reuse-related experience and self-efficacy) were major determinants. In addition, the findings suggested that more investigation is needed on nontechnical factors (i.e., prevailing attitudes and perceptions) that are barriers to software reuse.

Agresti in [33] investigated the developers' experiences and perceptions on software reuse in 2010. In this work, Agresti introduced the "4A" model which emphasized that for each organization to obtain any benefits from code reuse, four conditions must be met: Availability, Awareness, Accessibility, and Acceptability. Agresti, in his study, was more specific where the investigation done focused on code reuse. The findings from [33] revealed the greatest obstacle to reuse was shown to be awareness of reusable code and the developers' perceptions of its acceptability for use on their new projects. Interesting to note also, the developers felt that the complexity of old codes was the main reason why the codes were not reused.

In 2012, Chernak in [34] reported a survey conducted pertaining to the state of requirements reuse practice. The respondents came from the author's professional network across the globe. 82 responses (in which 60% of them resided in North America) were gathered during a six-month survey in 2010. Even though the respondents were aware of the reuse benefits, he found that poorly structured and badly maintained existing requirements were the main obstacles for adopting requirements reuse. He concluded that to improve reuse adoption, organizations should include refactoring existing requirements into a better structured model, maintaining a complete requirements model through releases, separating the stakeholder and product types, and imposing change impact analysis in their reuse practice.

The first three related works [28][29][30] focused on software reuse in general and the fourth one [32] focused on code reuse. The closest study that can relate to our work is from Yuri [34], which focused on reuse of requirements. However, we tried to replicate some of the items imposed in [31], [33], and [34] in our survey and put the context for understanding the practitioners' perceptions and experience in RR from the Malaysian perspective.

III. Research Model and Hypotheses

Easterbrook et al. in [35] outlined six steps in selecting empirical research method, beginning with clarifying research questions. The three research questions we are trying to answer include "what is the current state of RR practice in Malaysia?", "what are the factors than can influence RR to happen?", and "what are the reasons RR practice is not that common among software practitioners in Malaysia?" The following section describes the research model used in this study followed by our hypotheses.

A. Research Model

To answer the research questions, we modified the TAM variables used in [31] and used them with the "4A" factors presented by [33]. Some of the items posted in Yuri's survey [34] were also put into our survey.

In [31], Mellakord et al. mapped Behavioural Intentions (BI) towards four other variables: predetermined mindset, self-efficacy, perceived ease of use, and perceived usefulness. Mellakord rigorously evaluated the survey responses by using Structural Equations Modelling (SEM). The data were collected from 207 developers of 50 software companies in India. Their objective was to seek replies from the developers regarding software reuse in general, whereas in our case we are interested to seek responses from the software practitioners that deal specifically with requirements. Due to this, we obtained less responses as compared to [31], in which logically, not all developers deal with requirement documents. In some situations, a few practitioners we approached refused to answer our questionnaires because they had no experience dealing with requirement documents. Since less data were collected, SEM could not be used in our case. In addition, due to the time constraints in reporting this survey, we only report on the Behavioural Intention towards reuse and dropped two other variables used in [31]: predetermined mindset and perceived ease of use. The other two variables used in [31], self-efficacy and perceived ease of use, are considered as they also appeared in the Awareness factor in "4A" used by Agresti. The "4A" factors are used together with BI in constructing the survey hypotheses. Table 1 indicates the 4A factors adapted in our survey.

Table 1 4A Factors Adaptation

4A Factors	Used in Agresti [33]	Adaptation
Availability	Reusable artefacts	1. Support tools
Awareness	The existence of reusable artefacts	1. Self-efficacy 2. Reuse benefits
Accessibility	Ability to get the reusable artefacts	Ability to get the reusable requirements
Acceptability	Agreements on accepting the reusable artefacts in new projects	1. Conditions of the existing requirements. 2. Who decide to accept

Based on items posted in Yuri's survey [34], we have also incorporated seven external variables in constructing the hypotheses: Existing Requirements, Reuse v Develop new, Support Tools, Self-Efficacy, Job Performance, Team Productivity, and Maintenance Costs.

B. Hypotheses

We have formulated seven hypotheses based on the three research questions, the Behavioural Intention from TAM, the external variables, and the adaptation of 4A factors that appeared in Table 1. Table 2 indicates our research hypotheses.

Table 2 Research Hypotheses

No.	Hypotheses
Requirements reuse practice is influenced by:	
H1	The practitioners' *behavioural intentions* to reuse existing requirements;
H2	The *availability* of support tools for RR activities;
H3	The *awareness* of self-efficacy towards RR;
H4	The *awareness* of the easiness to reuse versus develop new requirements;
H5	The *awareness* of the impact of RR practice on job performance, team productivity, and maintenance costs;
H6	The *accessibility* to existing requirements; and
H7	The *acceptability* in terms of conditions of existing requirements.

For RR to occur, individual behavioural intention to reuse is the most important factor to consider (H1). Next, the conditions in the "4A" model must be satisfied in sequence. For example, firstly, if an organization is to practise reuse, they must provide the reuse support tools (H2). Adding to Agresti's view on Awareness of the existence of reusable artefacts, software practitioners must have the awareness of his or her ability (self-efficacy) to reuse the existing reusable requirements (H3 & H4). The developers or practitioners must also be aware of the benefits (usefulness) of RR towards their job performance, team productivity, and the effects on maintenance costs (H5). Subsequently, the developers or practitioners must be able to access the existing requirements from prior releases (H6). If the practitioners are unable to get the existing requirements, reuse is definitely impossible. After getting the access to reusable requirements, the conditions of the reusable requirements must be acceptable: they must be readable, complete, reasonably structured, and kept updated (H7).

IV. SURVEY CONSTRUCTIONS

Apart from Easterbrook's guide in selecting empirical methods for software engineering as appeared in [35], we also followed the guideline provided by Kitchenham and Pfleeger [36], [37], [16] and [17] on conducting survey research for software engineering and a guide by Kasunic from SEI Institute, Carnegie Mellon University on Designing Effective Survey [40].

Our survey consists of three parts:

- **Part A:** Demographic Information
- **Part B:** Requirements Reuse Perceptions and Experiences
- **Part C:** General Issues and Comments on RR (Open-Ended Questions)

We have adopted several questions from Mellakord [31] related to BI, used the "4A" category proposed by Agresti [33], and adopted some questions in Yuri [34] to suit our context of RR. Validity of the questionnaire items were tested on a pilot study. Results of the pilot test were used to check for Cronbach's alpha validity.

A. Pilot Testing

Forty-one sets of survey were distributed to the Software Engineering experts at 6 public universities in Malaysia that offer Software Engineering programmes. The responses from the pilot survey were tested for reliability in each of the test items. Each of the survey questions in Part B was tested in SPSS for internal consistency check by using Cronbach's α before being sent to the actual survey participants. Since the questionnaire used was modified from the survey conducted by [31], not many items yielded negative reliability values. When checked, two items were found to be negatively worded. Hence, the wordings were changed to positive wordings (i.e., *"decrease"* was changed to *"increase"* and *"is not important"* was changed to *"important"*) and the scores were reversed (i.e., from scale 1 to 7, the original score 6 was reversed to 2 and vice versa). The new scores were plugged in and retested in SPSS. Thus, the Cronbach's alpha is improved to 0.711, a more reliable value.

B. Actual Survey Data Collection

This survey is targeting to get responses from personnel who have experience dealing with requirement documents in software development. Our respondents consist of Software Engineers, Project Managers, Requirements Managers, Educators and Researchers in the Software Engineering areas, which we grouped in this survey as software practitioners. Although random selection is desired, this cannot be obtained. This is because no statistics were made available by MDec regarding the number of software practitioners in Malaysia that dealt with requirement documents during software development. Thus, a snowball sampling technique was used as the process to gather survey responses. In addition, a link to

a web-based survey questionnaire was posted on IT Professionals in Malaysia group on LinkedIn page, Malaysian Software Engineering Interest Group (MySeIG) page and Malaysian Research and Education Network (MyREN) page.

Demographic questions in Part A investigate the background of our respondents including:

a) Position in current job
b) Number of years in Requirements Engineering
c) Industry group that describes their organization
d) Size of development team
e) Requirements format used

Table 3 indicates the survey items used in Part B, which were derived from the hypotheses presented in Table 2.

Table 3 Hypotheses and Survey Items in Part B

No.	4A Factor	Survey Item (Rating 1 to 7)
H1	Behavioural Intention	1. I intend to increase my use of reusable requirements in the future development of application.
H2	Availability	2. My organization has appropriate *support tools* for: • developing reusable assets • managing reusable assets
*H3	Awareness	3. I feel reusing requirements requires a lot of mental effort (*self-efficacy*).
*H4	Awareness	4. It is easier for me to understand existing requirement documents as compared to developing new requirements (*reuse v develop new*).
*H5	Awareness	Reusing existing requirements: 5. improves my *job performance* 6. improves my *team productivity* 7. decreases *maintenance costs*
H6	Accessibility	8. Assuming I have access to *existing requirements*, I intend to use them when developing future applications. 9. Given that I have access to *existing requirements*, I predict that I would make use of them in developing future applications.
H7	Acceptability	10. It is impossible to reuse the existing requirements because the *existing requirements* developed in previous releases are incomplete or do not exist. 11. It is difficult to identify which requirements can be reused because the *existing requirements* are poorly structured. 12. It is difficult to use the existing requirements because the *existing requirements* are not kept updated.
*Note that there are three hypotheses on awareness: H3, H4, and H7. Observed from results presented in a previous study [33], awareness factor appears to be one of the major obstacles to code reuse. In addition, deeply held beliefs (including awareness) results in resistance from developers to reuse [31]. Thus, we strongly feel that it is worth to include the three hypotheses on awareness in our survey on RR.		

The following are additional questions imposed in the survey that were not particularly mapped to any hypothesis, but somehow important to be included:

a) Is there anyone who reuses requirements in the latest project?

b) If they reuse, what are the reasons for them to reuse:

- ✓ Reuse is systematically planned (SPL)
- ✓ Reuse just happens because the new project is very similar to the one completed before (ad hoc reuse)
- ✓ Reuse occurs because of maintaining previous release (software maintenance)

For each item in Part B, we use Likert-Scale 1 to 7 response options. Table 4 indicates the score rating.

Table 4 Likert Scales Rating

1	Strongly Disagree	Negative attitude (Disagree)
2	Disagree	
3	Slightly Disagree	
4	Neutral	Undecided
5	Slightly Agree	Positive attitude (Agree)
6	Agree	
7	Strongly Agree	

In the analysis, scores 1 to 3 indicate negative attitude (disagreement) while scores 5 to 7 indicate the respondents' positive attitude (agreement) to items in the questionnaires. The higher the score indicates, the more positive attitude (agreement) towards the item imposed. At the moment this paper is written, a total of 36 responses have been collected for analysis.

V. Results And Discussions

During the first 3 months (April 2013 until July 2013), we were able to collect 36 responses from the survey. Basic quantitative data analysis was done in Microsoft Excel and Frequencies Analysis in SPSS.

A. Demographic Information

Fig. 1, 2, 3 and Table 5 summarize the demographic information of survey respondents. The majority (52.7%) of our respondents were Software Engineers. Other respondents held various posts including Researchers and Educators in Software Engineering (22.2%), Project Managers (8.3%), System Analysts (5.6%), Technical Specialists (5.6%), Requirements Manager (2.8%), and Software Tester (2.8%).

When asking about experience in requirements engineering, more than half of the respondents have more than 1 year experience in Requirements Engineering (See Fig. 1).

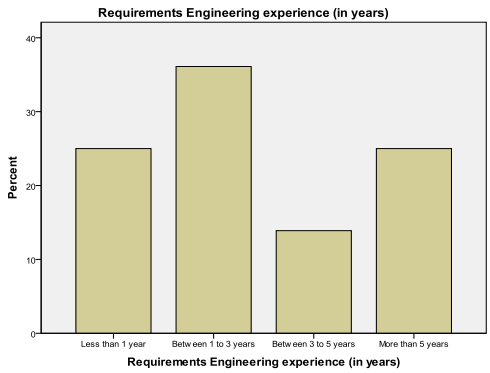

Fig. 1. Requirements Engineering Experience

Respondents came from various industries (Fig. 2), in which 36% came from Software Development House, 26%

from IT Consultancy and 15% from Education, Research & Development category.

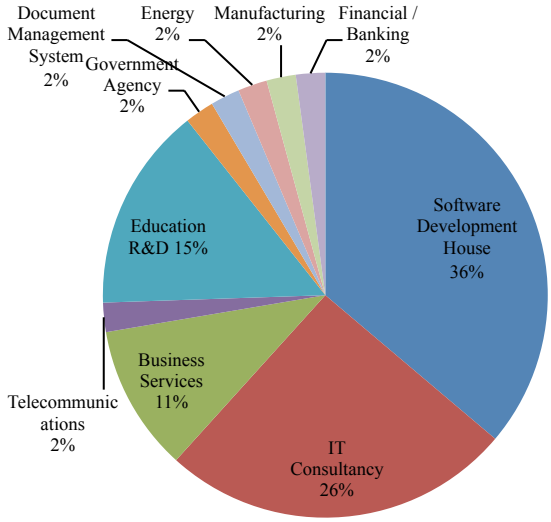

Fig. 2. Profile of Survey Respondents

Table 5 indicates that (50%) of the respondents worked in small development teams (between 1 to 5 people) and the remaining worked in various development teams sizes.

Table 5 Size of Development Teams

Team Sizes	Frequency	Percent
1 to 5 people	18	50.0
6 to 10 people	6	16.7
11 to 20 people	6	16.7
21 to 50 people	6	16.7
Total	36	100.0

Respondents were asked to categorize the requirements format used in the software development they were involved with. Requirements in the form of features (63.3%) and textual (63.89%) were among the famously used form of requirements (See Fig. 3). Features and textual requirements are usually represented in natural language. This finding indicates that natural language requirement is popular among the software practitioners in Malaysia. This trend is similar to a study conducted by Neil and LaPlante on the state of requirements engineering practice in [13], where developers mostly used requirements presented in natural language during software development. This is because documenting software requirement demands human interpretations, thus making natural language requirements more popular [14].

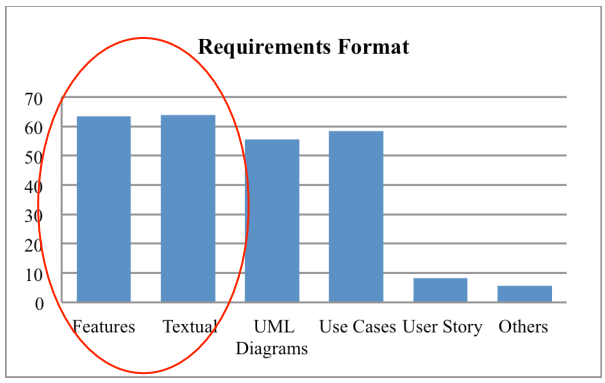

Fig. 3. Requirements Format Used by Respondents
**Note that respondents may choose more than one category, thus results totalling more than 100%*

B. *Perceptions and Experience in Requirements Reuse*

In this section, we will discuss the responses gathered on BI to RR and each of the "4A" factors used in hypotheses specified in Table 2. Data collected were re-coded in terms of agreement (Likert scores 5 to 7) and disagreement (Likert 1 to 3) as mentioned in Part B of Section IV of this paper. Throughout this section, undecided responses were discarded from the analysis.

1) Behavioural Intention

We first surveyed on the BI towards RR, H1. Results gathered indicate that 25 out of 36 respondents have the intention to reuse requirements in the future development as reported in Table 6.

Table 6 Behavioural Intention Towards RR

Responses	Frequency	Percent
Agree (Intention to reuse)	25	69.4
Disagree (No Intention to reuse)	11	30.6
Total	36	100.0

Therefore, we accept H1, RR practice is influenced by BI.

2) Availability

The first 4A factor in determining software practitioners' intentions to reuse is the availability of RR tools as specified in H2. For example, tools associated with managing and reusing requirements available in the current market are produced by BigLever, PureSystems, JAMA software, The Reuse Company, and some software extensions to DOORS by IBM. Most of the software mentioned are available for purchase, but not available for free download. The availability of support tools (fully automated or semi-automated) may

reduce the burden put on the requirements analysts while identifying the core and variant features for reuse in new product family development [4][41]. The availability of automatic tool support can also offer an order-of-magnitude savings over manual feature extraction for reuse [42], and obviously increase productivity when reuse is done systematically [43].

There were two survey questions pertaining to the availability of support tools: the first question seeks respondents who agreed that their organization provided support tools for developing reusable requirements, while the second seeks respondents who agreed that their organization provided support tools for managing reusable requirements. For the first question, 54.2% of the respondents did not agree that their organization provided support tools for developing reusable requirements. For the second question, 53.8% of the respondents did not agree that their organization provided support tools for managing reusable requirements. This result tells us that more than 50% of the respondents reported that no support tools are provided by their organization to aid RR activities (developing and managing).

Although support tools are important, our survey data indicated that most organizations did not provide it in the RR activities. In conjunction with this, our data also showed that most RR practitioners did not use any support tools during their last RR project: 27 out of 33 respondents (81.8%) did not use any support tool while requirements were reused in their latest project. Organizations did not provide support tools for RR, thus most of the practitioners did not use any tools in RR activities. We suspect this observation could have a relationship with the ad hoc RR practice (see details in Section 6). We have run a crosstab analysis to determine the pattern between respondents who did not use support tools in their latest RR project and compared it with the reason why requirements are reused in their latest project (SPL, ad hoc or software maintenance). As suspected, 16/27 of the respondents who did not use support tools were actually practising RR on ad hoc basis, 9/27 did not use support tools and reuse requirements for maintenance purposes, and 2/27 who did not use support tools were involved in systematic RR (SPL). In addition, only 4/6 who practised SPL use support tools in their latest RR project. This indicates that support tools were only used by many of the respondents who practised systematic RR (SPL), whereas those who were not involved in SPL did not use any support tools to aid their RR activities. Therefore, we strongly believe that our survey results provide a positive indication to accept H2: the availability of support tools can influence RR practice in SPL. Table 7 details out the crosstab analysis.

Table 7 Crosstabulation - Reason for Reuse v Using support tools in RR

		Requirements are reused in latest project because:			
		SPL	Just-happen (ad-hoc)	Maintaining prior release	Total
We use a support tool to assist our requirements reuse process	No	2	16	9	27
	Yes	4	2	0	6
Total		6	18	9	33

In relation to this, a few comments received in the open-ended section suggested that some of the practitioners still need to see a tool or framework for RR. So far, the closest RR tools they have seen were the UML diagrams, but not any specific tools that are capable to search and select existing textual-based requirements for reuse in new software development. This observation tells us that support tools are needed for SPL (systematic RR).

3) Awareness

Requirements reuse practice is influenced by the practitioners' awareness factors: H3 (awareness of self-efficacy), H4 (awareness of easiness to reuse versus develop new requirement), and H5 (awareness of the impact of RR). Fig. 4 below indicates the results pertaining to H3, H4, and H5.

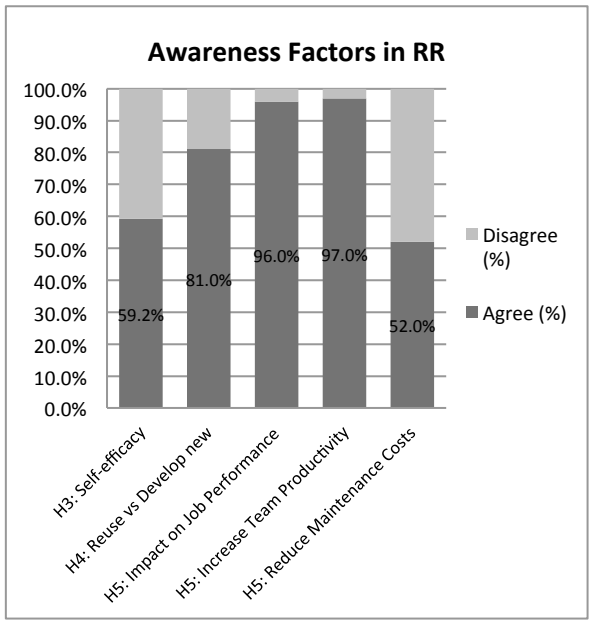

Fig. 4. Hypotheses on Awareness: H3, H4, and H5

When answering the awareness regarding self-efficacy, 16 out of 27 or 59.2% of the respondents agreed that RR requires a lot of mental effort. Although the respondents were aware of the difficulties to reuse, 81% agreed that it is easier to understand the reusable requirements as compared to

developing new requirement documents. This tells us that software practitioners were aware that to reuse is easier than to develop; however, reusing existing requirements will still need careful and rigorous thinking.

Regarding the awareness of the impact of RR, the majority of respondents agreed that RR provides good impact on their job performance and organization. 26 out of 27 (96%) of the respondents agreed that reuse can give positive impact on their job performance, 29 out of 30 (97%) agreed that RR increases their team productivity, and 15 out of 29 (52%) agreed that RR may reduce the maintenance costs at the later stage of software development.

Around 52% (13 out of 25 respondents) who intend to reuse requirements in the future development also agreed that reusing requirements requires a lot of mental effort. In addition to that, 17 out of 25 (68%) who intend to reuse requirements in the future development agreed that it is easier to understand reusable requirements as compared to developing new ones. Furthermore, from the 25 respondents who intend to practice RR in future development, 20 out of 21 agreed that RR will provide positive impact on their organization in terms of increasing their job performance, increasing team productivity (22 out of 23), and decreasing maintenance costs in the future (10 out of 23). This indicates that behavioural intention to reuse requirements in the future is related to the three reasons within the awareness factor. Therefore, we accept H3, H4, and H5; RR practice is influenced by the three awareness factors mentioned earlier.

4) Accessibility

H6 stated that RR practice is influenced by the *accessibility* to reusable requirements. This was tested in two survey questions. The results are presented in Table 7.

Table 7 H6 Accessibility factors in RR

		Acc1	Acc2
N	Valid	36	36
	Missing	0	0
Median		1.00	1.00
Mode		1(29)	1(29)

Acc1: *Assuming I had access to reusable requirements, I intend to use them when developing future applications*
Acc2: *Given that I have access to reusable requirements, I predict that I would make use of them when developing future applications*

From the findings tabulated in Table 7, mode 1 indicated that the respondents agreed to the statements in Acc1 and Acc2. 29 out of 31 intend to reuse and 29 out of 30 predict to reuse requirements if they have access to reusable requirements. This is reflected in the open-ended section as well, where the practitioners tend to refer back to existing documentation (functionality and templates) when developing requirements for new releases. Thus, we accept hypothesis H6, the accessibility factor.

5) Acceptability

H7 stated that the *acceptability* of reusable requirements influences RR practice. In a previous research conducted by [44] on requirements engineering problems in 63 software companies in Malaysia, the authors found out more than 70% of their respondents experienced problems related to requirements-process. These include inconsistent or changing requirements and incomplete requirements. With that in mind, we make an assumption that RR not being widely practised could be due to the conditions of existing requirements produced from RE activities, namely reusable requirements are incomplete, poorly structured (inconsistent) or do not exist.

We tested this assumption on the reuse experience in our respondents' organizations pertaining to the reason behind "Not Invented Here" syndrome for RR. In our survey, respondents were asked to respond to the three reasons why RR was not practised in their organizations. Fig. 5 summarizes the responses collected.

We found 33.3% of the respondents agreed that the requirements developed in previous releases were incomplete (or did not exist), so it is impossible to reuse them. Moreover, our findings revealed that 38.9% of our respondents agreed that existing requirements were poorly structured, and lastly 49.9% of the respondents thought that the existing requirements were not kept updated.

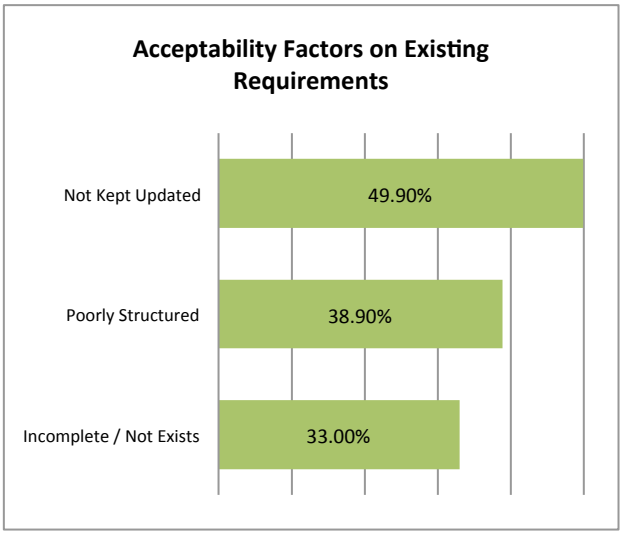

Fig. 5. Conditions of reusable requirements

Although most respondents have the intentions to practice RR in the future, the three reasons in the acceptability factors hinder RR to happen. When mapping the behavioural intention to acceptability factors, 8 out of 25 who intended to reuse requirements in the future development reported that old requirements did not exist in their organizations. In addition, 9 out of 25 reported to have poorly structured requirements and 13 out of 25 thought that old requirements were not kept updated. Therefore, although the intention to reuse exists, the conditions of reusable requirements (do not exist, not updated, and poorly structured) contribute to the reasons why RR is not widely practised in Malaysia, accepting the H7.

6) Additional Item

Additional items imposed in Section B include queries regarding the reasons why requirements were reused in the respondents' latest project as captured in Table 8. Only 19.4% of the respondents were involved in Systematic Reuse (SPL), while the majority (52.8%) of respondents reused requirements in an ad hoc manner.

Table 8 Reasons Requirements Were Reuse in Latest Project

Reasons:	Frequency	Percent
We are involved in SPL	7	19.4
Just happen the new project has similar requirements with the previously developed (ad hoc)	19	52.8
We are maintaining prior releases	10	27.8
Total	36	100

C. General Comments on RR (Open-Ended Question)

We provide a section for the respondents to write their general comments on RR practice in Malaysia. We tried to classify the comments into the "4A" category as depicted in Table 9.

Table 9 Mapping General Comments to 4A Factors

4A Factors	General Comments (Open-Ended)
Availability	Software engineering community has yet to see any RR tools or framework
Availability	Currently, my organization reuses 50% of older requirements. We make them as template for new development.
Availability	Need for an industry standard for exchanging and sharing requirements in repository.
Awareness	It is very important to educate developers on RR.
Awareness	Older requirements need to be revalidated prior to reuse and thus RR will not necessarily increase productivity.
Awareness	It is a good idea to use older requirements as it can help junior personnel involved in RE activities to learn.
Acceptability	RR in my organization is a case-by-case basis. Only those experienced will influence the decision-making of whether to reuse or not.

There were two comments that are related to the first 4A factor, Availability: to enable reuse, there is a need to have the RR tools, framework or the industry standard for exchanging and sharing requirements in repository. Tools will help to expedite reuse, while industry standard requirements repository will enhance reuse practice as practitioners can have a variety of requirements from a broad range of domain to choose from. Thus, time spent on RE activities may be reduced. The second important factor in 4A is Awareness. From the general comments section, we captured comments which are related to the awareness of the RR: the importance to educate developers with RR and RR can help juniors in the development team to learn. The last comments mentioned that only experienced personnel can make the decision whether to accept older requirements to be reused or not.

VI. Threats to Validity

At the time this paper is written, the online survey is still made available and open for more responses. We are aware that relying on the 36 responses obtained from the first three months of the data collection may introduce some threats to validity. However, this preliminary result provides us a general overview of the state of the practice in RR among the software practitioners in Malaysia.

As for the method to reach our respondents, using snowball sampling can introduce some threats to validity as well. This is because the survey link can be passed on to almost everyone and we are not able to control whether the respondents are the actual software practitioners who deal with requirement documents in their job.

In addition, the survey rating may be improved whereby the "Neutral" responses from the Likert Scale can be removed, thus respondents are forced to commit on their responses (to agree or disagree).

Lastly, the reliability of the survey results can be improved if we have more responses, thus a more rigorous statistical evaluation can be performed. In this paper, only frequencies analysis (median, mode, and manual cross-tabulation comparison) are performed against the data collected.

VII. Conclusions And Future Works

In this study, we have conducted a survey with two objectives: 1) to identify the factors influencing individual software practitioners in RR practice and 2) to assess the factors hindering software practitioners' from reusing requirements in software development, both in the Malaysian context.

We used the modified TAM variables from Mellakord et al. in [31] survey with "4A" factors [33] from Agresti's study on software reuse. We have tested hypotheses on seven factors around 4A factors that influence the RR practice.

From the survey results, we found seven factors that influenced the RR practice in Malaysia: behavioural intention, availability of support tools, awareness factors (self-efficacy, easiness to reuse versus developing new requirements and impact of RR), accessibility to reusable requirements, and the acceptability conditions of reusable requirements.

Even though BI to practice RR was observed to be highly agreed by most of the respondents, our survey indicates that the RR practice is not widely practised in Malaysia mainly due to three impediments: unavailability of RR tools, unacceptable conditions of requirements to be reused, and the lack of RR education or guidelines provided.

In addition, our study uncovers the status of SPL practice in Malaysia. We found very limited systematic reuse practice among the software practitioners in Malaysia. Most reuse experience captured in our study appears to be on ad hoc basis. Pertaining to intention to reuse existing requirements, although reusable requirements were available and accessible, respondents found that the reusable requirements were incomplete, poorly structured or not kept updated; thus making them unacceptable to reuse. These two findings relating to

reuse of requirements synchronize well with findings reported in [34].

This study also reveals that textual-based or natural language is the most popular format of requirements used in software development, reconfirming the statement made in earlier research appeared in [13]. However, according to the respondents, no tool or framework is available except the requirements gathering process or tools like UML which were used in ad hoc RR. Thus, in the near future, we are looking into developing a prototype to demonstrate how features can be selected from natural language requirement documents to support reuse in a similar software product development, SPL. This is hoped to contribute to demonstrating and promoting the systematic RR among the software engineering community in Malaysia.

ACKNOWLEDGEMENT

This work is supported by University of Malaya Research Grant, RG094/12ICT and FP050-2013A.

REFERENCES

[1] K. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson, "Feature Oriented Domain Analysis (FODA) Feasibility Study," Pittsburgh, PA, 1990.

[2] M. Eriksson, J. Börstler, and K. Borg, "Managing requirements specifications for product lines – An approach and industry case study," *J. Syst. Softw.*, vol. 82, no. 3, pp. 435–447, Mar. 2009.

[3] A. Mahmoud and N. Niu, "An experimental investigation of reusable requirements retrieval," *2010 IEEE Int. Conf. Inf. Reuse Integr.*, pp. 330–335, Aug. 2010.

[4] N. Weston, R. Chitchyan, and A. Rashid, "A Framework for Constructing Semantically Composable Feature Models from Natural Language Requirements," in *Software Product Lines Conference*, 2009, pp. 211–220.

[5] L. Zong-yong, W. Zhi-xue, Y. Ying-ying, W. Yue, and L. Ying, "Towards a Multiple Ontology Framework for Requirements Elicitation and Reuse," *31st Annu. Int. Comput. Softw. Appl. Conf. - Vol. 1- (COMPSAC 2007)*, no. Compsac, pp. 189–195, Jul. 2007.

[6] E. K. Karatas, "An Ontology Based Approach to Requirements Reuse Problem in Software Product Lines," Middle East Technical University, 2012.

[7] A. N. Lindoso and R. Girardi, "The SRAMO Technique for Analysis and Reuse of Requirements in Multi-agent Application Engineering," pp. 1–10.

[8] J. Dehlinger and R. R. Lutz, "Supporting Requirements Reuse in Multi-Agent System Product Line Design and Evolution," in *ICSM 2008*, 2008, pp. 207–216.

[9] H. Hara, S. Aijitat, and K. Sugawara, "Reusable Software Components based on an Agent Model," pp. 447–452, 2000.

[10] M. Wooldridge, "The Gaia Methodology for Agent-Oriented Analysis and Design," *ACM Trans. Softw. Eng. Methodol.*, vol. 12, pp. 285–312, 2000.

[11] E. J. Stierna and N. C. Rowe, "Applying information-retrieval methods to software reuse: a case study," *Inf. Process. Manag.*, vol. 39, no. 1, pp. 67–74, Jan. 2003.

[12] J. Yan, S. Yan, N. Liu, and Z. Chen, "Straightforward Feature Selection for Scalable Latent Semantic Indexing," pp. 1160–1171.

[13] C. J. Neill and P. A. Laplante, "Requirements Engineering : The State of the Practice," *IEEE Softw.*, vol. 20, no. 6, pp. 40–45, 2003.

[14] T. C. Oliveira, P. Alencar, and D. Cowan, "ReuseTool—An extensible tool support for object-oriented framework reuse," *J. Syst. Softw.*, vol. 84, no. 12, pp. 2234–2252, Dec. 2011.

[15] L. B. Lisboa, V. C. Garcia, D. Lucrédio, E. S. de Almeida, S. R. de Lemos Meira, and R. P. de Mattos Fortes, "A systematic review of domain analysis tools," *Inf. Softw. Technol.*, vol. 52, no. 1, pp. 1–13, Jan. 2010.

[16] J. Nicolás and A. Toval, "On the generation of requirements specifications from software engineering models: A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 9, pp. 1291–1307, Sep. 2009.

[17] R. T. Kolagari and M. Reiser, "Reusing Requirements : The Need for Extended Variability Models," pp. 129–143, 2007.

[18] E. Boutkova and F. Houdek, "Semi-Automatic Identification of Features in Requirement Specifications," in *2011 IEEE 19th International Requirements Engineering Conference*, 2011, pp. 313–318.

[19] K. S. Barber, T. J. Graser, P. S. Grisham, and S. R. Jemigan, "Requirements evolution and reuse using the systems engineering process activities (sepa)," *Australas. J. Inf. Syst.*, vol. 6, no. 2, pp. 75–97, 2007.

[20] A. Toval, J. Nicolás, B. Moros, and F. García, "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," *Requir. Eng.*, vol. 6, no. 4, pp. 205–219, Jan. 2002.

[21] V. Alves, N. Niu, C. Alves, and G. Valença, "Requirements engineering for software product lines: A systematic literature review," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 806–820, Aug. 2010.

[22] N. Niu and S. Easterbrook, "On-Demand Cluster Analysis for Product Line Functional Requirements," *2008 12th Int. Softw. Prod. Line Conf.*, pp. 87–96, Sep. 2008.

[23] N. Niu and S. Easterbrook, "Extracting and Modeling Product Line Functional Requirements," *2008 16th IEEE Int. Requir. Eng. Conf.*, pp. 155–164, Sep. 2008.

[24] K. Chen, W. Zhang, H. Zhao, and H. Mei, "An approach to constructing feature models based on requirements clustering," *13th IEEE Int. Conf. Requir. Eng.*, pp. 31–40, 2005.

[25] V. Alves, C. Schwanninger, L. Barbosa, A. Rashid, P. Sawyer, P. Rayson, C. Pohl, and A. Rummler, "An Exploratory Study of Information Retrieval Techniques in Domain Analysis," in *2008 12th International Software Product Line Conference*, 2008, pp. 67–76.

[26] V. Alves, C. Schwanninger, L. Barbosa, A. Rashid, P. Sawyer, P. Rayson, C. Pohl, and A. Rummler, "An Exploratory Study of Information Retrieval Techniques in Domain Analysis," *2008 12th Int. Softw. Prod. Line Conf.*, pp. 67–76, Sep. 2008.

[27] A. Von Knethen, B. Paech, F. Kiedaisch, F. Houdek, D.-Kaiserslautern, D.- Ulm, and D. Ag, "Systematic Requirements Recycling through Abstraction and Traceability," in *Requirements Engineering*, 2002, pp. 273–281.

[28] M. Mannion and H. Kaindl, "Using Parameters and Discriminants for Product Line Requirements," *Syst. Eng.*, vol. 11, no. 1, pp. 61–80, 2007.

[29] W. B. Frakes and C. J. Fox, "Sixteen Questions About Software Reuse," *Commun. ACM*, vol. 38, no. 6, pp. 75–87, 1995.

[30] O. P. N. Slyngstad, A. Gupta, R. Conradi, P. Mohagheghi, H. Rønneberg, and E. Landre, "An Empirical Study of Developers Views on Software Reuse in Statoil ASA," no. Idi, pp. 242–251, 2006.

[31] V. Mellarkod, R. Appan, D. R. Jones, and K. Sherif, "A multi-level analysis of factors affecting software developers' intention to reuse software assets: An empirical investigation," *Inf. Manag.*, vol. 44, no. 7, pp. 613–625, Oct. 2007.

[32] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Manage. Sci.*, vol. 35, no. 8, pp. 982–1003, 1989.

[33] W. W. Agresti, "Software Reuse: Developers' Experiences and Perceptions," *J. Softw. Eng. Appl.*, vol. 04, no. 01, pp. 48–58, 2011.

[34] Y. Chernak, "Requirements Reuse: The State of the Practice," in *IEEE International Conference on Software Science, Technology and Engineering*, 2012, pp. 46–53.

[35] S. Easterbrook, J. Singer, M. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," in in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. K. Sjøberg, Eds. Springer London, 2008, pp. 285–311.

[36] B. A. Kitchenham and S. Software, "Principles of Survey Research Part 2 : Designing a Survey," *ACM SIGSOFT*, vol. 27, no. 1, pp. 18–20, 2002.

[37] B. A. Kitchenham and S. L. Pfleeger, "Principles of Survey Research Part 3 : Constructing a Survey Instrument," *ACM SIGSOFT*, vol. 27, no. 2, pp. 20–24, 2002.

[38] B. Kitchenham, "Principles of Survey Research Part 4 : Questionnaire Evaluation," *ACM SOGSOFT*, vol. 27, no. 3, pp. 20–23, 2002.

[39] B. Kitchenham and S. L. Pfleeger, "Principles of Survey Research Part 5 : Populations and Samples," *ACM SIGSOFT*, vol. 27, no. 5, pp. 17–20, 2002.

[40] M. Kasunic, "Designing an Effective Survey," 2005.

[41] K. Kumaki, R. Tsuchiya, H. Washizaki, and Y. Fukazawa, "Supporting commonality and variability analysis of requirements and structural models," *Proc. 16th Int. Softw. Prod. Line Conf. - SPLC '12 -volume 1*, p. 115, 2012.

[42] N. Niu, J. Savolainen, Z. Niu, M. Jin, and J.-R. C. Cheng, "A Systems Approach to Product Line Requirements Reuse," *IEEE Syst. J.*, pp. 1–10, 2013.

[43] F. Barreto, V. Benitti, and R. Cezario, "Evaluation of a Systematic Approach to Requirements Reuse," *J. Univers. Comput. Sci.*, vol. 19, no. 2, pp. 254–280, 2013.

[44] B. Solemon, S. Shahibudin, and A. A. Abdul Ghani, "Requirements Engineering Problems in 63 Software Companies in Malaysia Badariah Solemon," in *International Symposium on Information Technology, ITSim 2008*, 2008, pp. 1–6.