Issues in Requirements Specification in Malaysia's Public Sector: An Evidence from a Semi-Structured Survey and a Static Analysis

Mohd Firdaus Zahrin[1], Mohd Hafeez Osman[2]*, Alfian Abdul Halin[3], Sa'adah Hassan[4], Azlena Haron[5]
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Malaysia[1, 2, 3, 4]
National Administrative Institute, Public Service Department of Malaysia[5]

Abstract—**Requirement specifications (RS) are essential and fundamental artefacts in system development. RS is the primary reference in software development and is commonly written in natural language. Bad requirement quality, such as requirement smells, may lead to project delay, cost overrun, and failure. Focusing on requirement quality in the Malaysian government, this paper investigates the methods for preparing Malay RS and personnel competencies to identify the root cause of this issue. We conducted semi-structured interviews that involved 17 respondents from eight critical Malaysian public sector agencies. This study found that ambiguity, incompleteness, and inconsistency are the top three requirement smells that cause project delays and failures. Furthermore, based on our static analysis, we collected the initial Malay RS documents from various Malaysian public sector agencies; we found that 30% of the RS were ambiguous. Our analysis also found that respondents with more than 10 years of experience could manually identify the smells in RS. Most respondents chose the Public Sector Application Systems Engineering (KRISA) handbook as a guideline for preparing Malay RS documents. Respondents acknowledged a correlation between the quality of RS and project delays and failures.**

Keywords—Ambiguity; requirements engineering; requirement smell; requirement specification; semi-structured interview

I. Introduction

Software Requirement Specification (SRS) is a document that details the behaviour of a system. It describes the functionality and non-functionality of the software to meet the requirements of all relevant stakeholders (business, users, and software). RS impacts entire software development project stages [1] as it may specify the Business Requirements Specification (BRS), User Requirements Specification (URS), and Software Requirements Specification (SRS) that define the expectation of the stakeholder of a system to be developed. RS is commonly expressed in Natural language [2], which is exposed to requirement smell [3] such as ambiguity, inconsistency, incompleteness, etc. [4]. The requirement smells impact the requirements' quality which may lead to low product quality.

In Malaysia, the public sector has funded a massive amount of money to digitise its systems. A lot of systems have been developed to realise the e-government initiative. However, from our initial observation, some system development projects faced problems such as project delays and low-quality product quality. We may assume that RS can be one of the causes of this problem. Based on a survey, 50% of requirement engineers were unaware of ambiguous Malay RS [5]. Zahid et al. [6] have highlighted those smells in RS cause roughly 70% of the project to fail. A lack of precise RS is one of the causes of project failure [7]. Unintentional ambiguity in natural language requirements leads to diverse implementations later in software development [8]. Based on Sommerville [9], "Clients of systems often struggle to translate their desires into requirements," resulting in smells and general terms in the requirements [4]. According to Iqbal et al. [10], "Ambiguous and generalised RS add significantly to project time and cost." Quality RS is determined by comprehending and managing requirements correctly [11]. Furthermore, Rios et al. [12] mentioned that based on survey results, 53% of respondents (software developers) reported that document debt is associated with requirement issues. These previous studies show that requirement smells can cause project delays or failures.

These issues motivate us to investigate the requirement smells and other problems in Malay RS. We focus on Malaysian government projects since most of the RS in the Malaysian government usually use the Malay language.

This paper aims to investigate the preparation of Malay RS and identify the issues that can result in project delays and project failures (incomplete or unfinished). This paper focus on four (4) perspectives (in four (4) research questions (RQ)): The Human factor (RQ1), the Communication factor (RQ2), the Process and Procedure (RQ3), and the Issues in RS (RQ4). We formulate the RQ (as shown in Table I) based on the Ab Aziz [13] guidelines.

This study contributes to the following:

1) We discover factors influencing the requirement smells in Malay RS.

2) We reveal the relation between requirement smells detection and software developer experience.

The remaining sections of this work are organised as follows. The Related Works section describes the relevant studies. The Methodology section reports the approach used to conduct the research. The Findings section explains the result. We discuss the findings in the Discussion section, and the Conclusion section summarises the paper.

*Corresponding Author.

TABLE I. RESEARCH QUESTIONS AND RESEARCH OBJECTIVES

Problem Statement	Research Question (RQ)	Research Objective (RO)
The requirement smells in Malay RS contribute to project delays and project failures.	RQ1: How competent are Malaysian public sector requirement engineers and software developers?	RO1: To investigate the competency level of requirement engineers and software developers in the Malaysian public sector.
	RQ2: What is the most used language in RS?	RO2: To identify the most language used in RS.
	RQ3: How is RS prepared and verified?	RO3: To identify the methods used to prepare and verify the RS.
	RQ4: What are the issues in RS?	RO4: To explore and identify the RS smells.

II. RELATED WORKS

We explored credible sources and found articles that are close to our study are the following:

A. Survey on Agile Requirements Engineering (RE) Practices

Barata et al. [14] studied the agile requirements in practice that focused on how professionals view the significance of requirements in an Agile methodologies-based software development project. A survey of 46 Brazilian software development experts asked about methods for collecting and expressing Agile requirements and their features, benefits, and challenges. The authors studied respondents' experience, viewpoints on collection methods, and Agile requirements specifications. On the contrary, we focused on some factors, i.e., investigating the issues in RS, methods for preparing the RS, RS verification and validation approaches, and requirement engineers' competencies.

B. Survey on Requirement Smells Among IT Practitioners

Lenarduzzi et al. [15] surveyed requirement engineers to understand the theoretical and practical perceptions of the harmfulness of requirement smells and to compare these perceptions with one another. A precise and validated approach for reducing issues during requirement elicitation is proposed to avoid introducing requirement smells. The authors focused the survey among requirement engineers on requirement elicitation processes to prevent the requirement smells. In contrast, our survey involved requirement engineers and software developers regarding the smells commonly found in requirements documents that could delay and fail software development projects.

Rios et al. [12] surveyed to obtain feedback from a software engineer on the factors leading to documentation debt, the effects of this problem, and potential solutions. The term document debt (DD) describes issues with software project documentation, namely the search for missing, inconsistent, out-of-date, or inadequate documentation [16]. These surveys (questionnaires and an interview-based case study) involved 39 practitioners from replications in Brazil, Chile, Colombia, and the United States. The fourth research question, "Which phase of the software development life cycle is most affected by the presence of DD?" reveals a substantial correlation between DD and requirements problems. In their examples of DD, around 53% of participants highlighted requirements issues. Examples include: "Lack of clarity and precision in the formulation of requirements" and "Needing to construct unspecified code because the requirement was not addressed in the documentation." The researchers focused on identifying the factors that led to DD based on the software development life cycle. In the way of comparison, we focused on the requirements phase. We surveyed the strategies in preparing Malay RS to determine the cause of requirement smells based on IEEE [4] quality attributes.

Within the software development life cycle context, Ahmad et al. [17] published the results of a survey that aimed to ascertain the relative importance of software requirements defects. Using questionnaires, the authors surveyed Malaysian IT professionals from diverse businesses (public, private, and software houses) and job titles (business analyst, system analyst, software engineer, etc.). According to the result, the requirement has defects, such as missing, incorrect, inconsistent, ambiguous, etc. Ahmad et al. [17] concentrated on the survey employing instruments such as questionnaires. In contrast, we conducted a semi-structured interview to investigate the smells in Malay RS documents in Malaysia's public sector agencies. Ahmad et al. [17] did not mention the language of the requirements. We assumed the authors selected the English requirements for their study. In this paper, we highlighted the most significant requirement smells.

C. Survey on Formalising System Requirements and Validation

Mokos and Katsaros [18] surveyed advancement in formalising and early validating system requirements. Several industrial research projects have provided valuable experience in pattern languages and formal property derivation. RS can use domain ontologies to identify missing information, inconsistencies, and under-specification. Our study differs because the author did not specify the survey method, whereas we used the semi-structured interview method. Mokos and Katsaros [18] also reviewed design paradigms not covered by our research.

D. Survey on Industrial IT Professionals' Awareness of Malay SRS

Haron and Abdul Ghani [5] surveyed IT professionals about their perception of ambiguity in the Malay SRS. The survey shows that IT professionals, especially in SRS, tend to overlook ambiguity. More than 50% of the respondents were unaware of the occurrence. The need for a tool to help solve the problem is undeniable. The difference with our study is; that the author sent a set of Malay SRS to IT professionals to assess their understanding of the functional specification. Meanwhile, we used Likert-scale and open-ended questions to gather software developers' understanding (semi-structured interview).

E. Survey on RE in Practice

Aguilar et al. [19] investigated the impact of RE practice in 16 small-sized software firms through a survey in Sinaloa, Mexico. The degree of relationship between each variable was determined using Pearson correlation analysis. The identified variables, i.e., company location, the scope of coverage, number of workers, etc. The survey found no strong correlation between the seven variables analysed. The significant issues to address; are i) Ineffective client-software development team communication can harm the RS; ii) Ad-hoc RE practices that are hard to track and possibly lost (untraceable) because respondents think RE is not essential for small firms. Our study differs; Aguilar et al. [19] concentrated on issues related to RE practice in the organisation through questionnaires, while our research focused on smells and issues in RS obtained from semi-structured interviews.

Ilyas et al. [20] surveyed Pakistani software companies' current requirement process practices for identifying issues. 10% of companies are unaware of the value of standards in product development. The most critical activity in software development is gathering system requirements. This step is crucial to understanding the user's needs. Our study differs because the author used questionnaire methods, and Ilyas et al. [20] focused on organisational issues. Çamoğlu and Kandemir [21] investigated the software RE processes in Turkish firms (energy, finance, and telecommunication sector). This study identified common issues and problems in business analysis. Insufficiently defined requirements and demands are the leading cause of process failure in all sectors. There is no standardisation of RS documentation. The most commonly used RS documentation tools are Microsoft Word and Excel. The authors focused on high-level RE issues while we concentrated on RS issues. Çamoğlu and Kandemir [21] collected data from many participants via questionnaires, but we interviewed a few for more in-depth feedback.

F. Survey on Techniques Suggested by Industrial Standards

Jarzębowicz and Połocka [22] surveyed requirements documentation techniques in various software project contexts in Poland IT firms. The survey asked 42 Polish IT professionals to choose strategies for multiple projects. After the survey, two interviews with business analysts were conducted to interpret the results. Our study differs because Jarzębowicz and Połocka [22] focused on RS and documentation techniques while we concentrated on problems and requirement smells in RS. The authors used questionnaires and interviews, but we used semi-structured interviews.

G. Quality Assurance in RE

Noorin and Sirshar [23] organised a survey on quality assurance in RE. The authors focused on analysing quality parameters that ensure the requirements are met. The authors discovered that models like LaQuSo Software Product Certification Model, Neural Network, Case-Based Reasoning, and others are used to check other quality parameters like correctness and completeness. In addition, analysing quality characteristics improve requirement quality. Our study focused on RS issues gathered through semi-structured interviews, while the authors surveyed models and quality attributes to strengthen requirements.

H. Factors of Projects Failure in Terms of RE Processes

Hussain and Mkpojiogu [24] investigated the success or failure factors of software development projects related to RE processes using secondary data analysis. As can be seen, poor requirements processes lead to software project challenges and failures. However, a reasonable requirements collection and process contributes to software project success. The authors advise software project planners, engineers, and managers to include adequate RE processes in all software development projects to avoid failures and challenges. It differs from our study; the authors concentrated on causes related to RE processes that influence the software development project failure or success. In contrast, we focused on requirement smells contributing to project delays and loss.

III. Methodology

This study is conducted to understand and observe RS preparation in the Malaysian public sector. We followed Ab Aziz's [13] guidelines to perform the study through the semi-structured interview. Overall, there are nine (9) steps to conduct the study, i.e., identify respondents, develop interview questions, conduct pre-interview, send invitations to respondents, interview, collect requirement documents, analyse interview results, analyse requirement documents, and conclude the findings. Fig. 1 illustrates the research methodology.

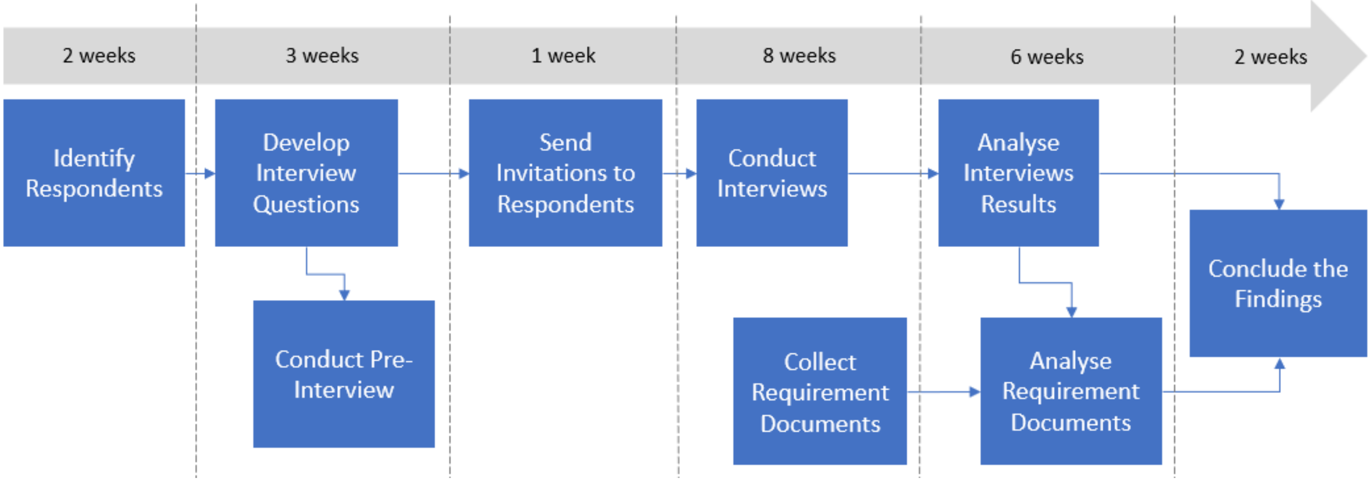
2 weeks
3 weeks
1 week
8 weeks
6 weeks
2 weeks
Identify Respondents
Develop Interview Questions
Conduct Pre-Interview
Send Invitations to Respondents
Conduct Interviews
Collect Requirement Documents
Analyse Interviews Results
Analyse Requirement Documents
Conclude the Findings

Fig. 1. Methodology of the Study.

A. Identify Respondents

First, we emailed selected agencies to nominate 15 respondents based on the criteria specified in Table II. Fortunately, we received feedback from various critical public sector agencies involving 17 respondents.

TABLE II. RESPONDENTS' CRITERIA SELECTION

Roles:	**Requirement Engineer**	**Software Developer**
The critical domain of Government agencies	Yes	
Expertise	Attended any RS course	N/A
Experience	At least two (2) years	
Involve in RS	Preparing the RS	Developing the software based on the RS

B. Develop Interview Questions

Indeed, questions are listed as a guide to ensure the smooth running of the interview. The developed semi-structured interview questions are based on RQs in the Introduction section. There are 16 main questions; seven open-ended questions, two Likert-scale, and six follow-up questions. An expert has validated these questions. The details of the interview questions are in Table III.

TABLE III. INTERVIEW QUESTIONS

	Question
RQ1 - Identify the competencies	
1)	How many years of service?
2)	How many years of service as a requirement engineer/ software developer?
3)	Have you attended any course related to preparing the requirements specification? Have you ever attended a professional course such as CPRE/ academic at the University (Requirement Engineering subject)?
RQ1 - Level of understanding of the requirements specification (software developer perspective)	
4)	Do you understand each of the requirement specifications? If not, why?
5)	Overall, state your understanding of the requirements specification used as a reference in developing the system. (1 - Do not understand to 5 - Understand) – Likert scales
RQ2 - The most used language in requirements specification	
6)	What are the languages used in preparing the requirements specification?
7)	How much percentage of requirements specifications are prepared in specified languages?
8)	Is there any policy for using a specific language for the requirements document?
9)	What is your preferred language for requirements specifications? And why?
RQ3 - The methods used to prepare and verify the requirements specification	
10)	What are the templates used in preparing the requirements specification? What are these templates used for?
11)	Is there any verification for the requirements specification? If not, why? If yes, how is the verification conducted? What are the verifications quality attributes?
12)	What are the tools used for requirements verification?
13)	Do you need a tool to check the quality of the requirements specification? (e.g., Checking the ambiguity, inconsistency, and incompleteness in the requirements specification) If yes, what are the specific features that you require? (e.g., syntax-based checking, template-based checking)
RQ4 - Identify the issues in requirements specification (software developer perspective)	
14)	What are the issues that frequently occur in understanding the requirements specification? How do you solve the problems?
15)	How much do you agree that the problem in the requirements causes project delays? (1 - Strongly disagree to 5 - Strongly agree) – Likert scales
16)	Choose the top three (3) smells commonly found in the RS that may contribute to project delays/ failures.

***Quality Attributes**	***Description***
Unambiguity	The requirement is expressed so that it can be interpreted only one way. The requirement is defined and easy to understand.
Appropriate	The specific purpose and detail of the requirement are suitable to the degree of the body to which it refers (level of abstraction appropriate to the entity's class).
Complete	The requirement adequately briefs the necessary capability, characteristic, constraint or quality factor in meeting the entity's need without needing other information to understand the requirement.
Verifiable	The requirement is structured and worded so that its realisation can be proven (verified) to the customer's satisfaction at the requirements level. Verifiability is enhanced when the requirement is measurable.
Necessary	The requirement defines an essential capability, characteristic, constraint or quality factor.
Correct	The requirement accurately represents the entity's need from which it was transformed.
Singular	The requirement states a single capability, characteristic, constraint or quality factor.
Feasible	The requirement can be realised within system constraints (e.g., cost, schedule, technical) with acceptable risk.

* IEEE [4] quality attributes

C. Conduct Pre-Interview

Before conducting semi-structured interviews, pre-interviews were performed to validate the questions developed. Two experienced system analysts from the Malaysian public sector agency (selected based on the criteria in Table II) participated in the preliminary interviews.

D. Send Invitations to Respondents

Afterwards, invitations to respondents are made via email and attached to an invitation letter. When the respondents agree to participate, the date and time are set based on the respondents' availability.

E. Conduct Interviews

Semi-structured interviews are conducted via online video communication for one hour each. This interview involves eight critical agencies. Due to confidentiality, we could not name the agencies. Table IV lists the study's respondents.

TABLE IV. LIST OF RESPONDENTS

Respondent (R)	Position	Grade	Working Experience	Experience with RS
R1	REng	44	11 years	9 years
R2	REng	48	18 years	3 years
R3	REng	44	14 years	10 years
R4	REng	41	9 years	6 years
R5	REng	44	14 years	6 years
R6	REng	44	20 years	10 years
R7	REng	48	17 years	12 years
R8	REng	44	11 years	6 years
R9	REng	44	13 years	10 years
R10	SD	29	8 years	8 years
R11	SD	29	12 years	10 years
R12	SD	32	13 years	13 years
R13	SD	32	15 years	13 years
R14	SD	29	11 years	10 years
R15	SD	29	6 years	2 years
R16	SD	32	18 years	18 years
R17	SD	32	13 years	2 years

Requirement Engineer (REng), Software Developer (SD)

F. Analyse Interview Results

This paper employed data visualisation and descriptive-analytic techniques by Regnell et al. [25] to analyse and synthesise the data collected following the RQs presented in the Introduction section.

G. Collect Requirement Documents

We gathered the Malay RS documents to confirm the existence of the issues raised by respondents. We officially request the Malay RS documents from 10 critical public sector agencies (11 domains). Due to confidentiality and requested by respective agencies, we could not reveal the agencies. After receiving the agencies' approval, we collected 18 documents, including the 12 BRS/URS and six SRS. These RS documents were sent to us via email in Microsoft Word and Adobe PDF format. Other than that, we received some RS documents in hard copies. Table V lists the domain of the collected Malay RS documents.

H. Analyse Requirement Documents

We identify and eliminate the non-RS. We chose and extracted the textual RS into SQL format (repository). Next, we cleaned the textual-based RS using the Rapid Miner [26]. The data cleaning process comprises identifying, addressing missing values, and manually validating Malay spelling. We manually labelled the class based on the most significant requirement smell. Two Malay requirement specialists confirmed the class label. This paper employed the statistical descriptive-analysis method by Christopher [27] to analyse the percentage of the most significant requirement smell mentioned by respondents.

TABLE V. DOMAIN OF COLLECTED MALAY RS

Domain	No. of RS
Business	65
Career	6
Customer Services	137
Funds/ Finance	892
Human Resources	76
General Administration	78
Miscellaneous	21
Survey	107
Trade	89
Training	49
Welfare	15
Total RS:	1535

I. Conclude the Findings

We conclude the results in the Findings section and discuss the insights in the Discussion section.

IV. FINDINGS

This section presents the key findings that answered the RQs in the Introduction section through the semi-structured interviews. Section A corresponds to the competency level of respondents (RQ1). Section B reports the languages employed by RS (RQ2). Section C corresponds to methods used to prepare and verify the RS (RQ3). Section D presents the issues in RS (RQ4).

A. Human Factor (RQ1)

1) Requirement engineers: This study investigates the competency level of requirement engineers in Malaysia's public sector. We found the respondent's minimum and maximum years of service were nine and 20. The minimum and maximum years of involvement in requirement engineering were between three and 12 years. Fig. 2 illustrates the years served as a requirement engineer. 78% of requirement engineers have attended various courses related to RS. The courses, i.e., Certified Professional Requirement Engineer (CPRE) [28], took a subject at university, boot camp, and internal courses organised by respective agencies. Hence, most of the requirement engineers in Malaysia's public sector are competent and well-equipped with training. This study also shows that we have selected trained and experienced requirement engineers.

2) Software developers: We studied the competency level of software developers in Malaysia's public sector. As a result, the respondent's minimum and maximum years of service were six and 18 years of experience. Fig. 3 shows the years served as a software developer.

3) Level of Understanding of the RS: This study determined the understanding of the SRS provided by the software engineer from the software developer's standpoint. Therefore, we require experienced software developers with a good understanding of SRS for software development. Fig. 4 illustrates that 86% of the respondents understood the RS given to them by the requirement engineer during the software development. However, 14% of respondents sometimes did not understand RS provided by their requirement engineer. Several respondents mentioned using flowcharts and mock interfaces, not RS, as a reference for system development. Fig. 5 shows that 57% of respondents understood the overall RS used as a reference in developing the system, while 43% of the respondents sometimes did not understand. The top three typical issues found in RS were ambiguity, incomplete and incorrect. Other than that, RS was constantly changing specifically for in-house development. This condition makes it difficult for software developers to expand the system and contributes to project delays. Most of the resolution of these issues is meeting with the project team and discussing the impact analysis and feasibility of the project with requirement engineers and clients.

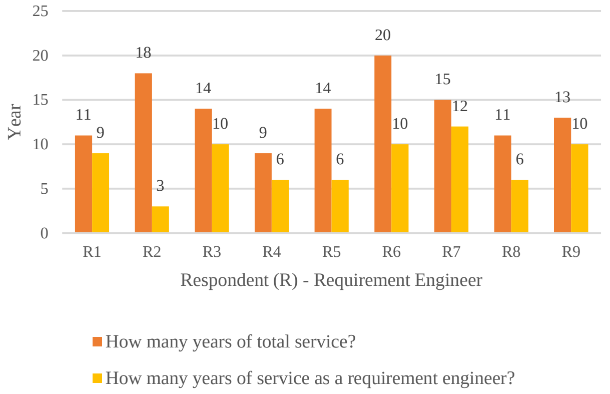

Fig. 2. Number of Years Served as Requirement Engineers.

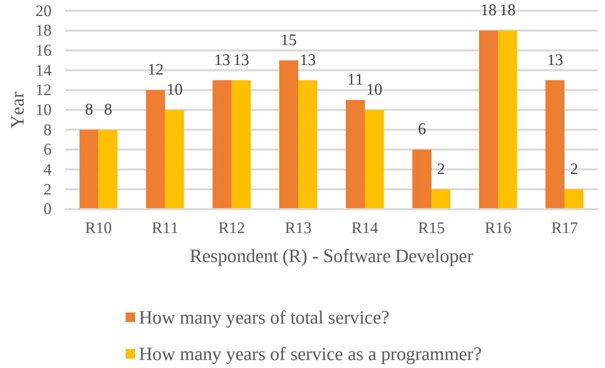

Fig. 3. Number of Years Served as Software Developers.

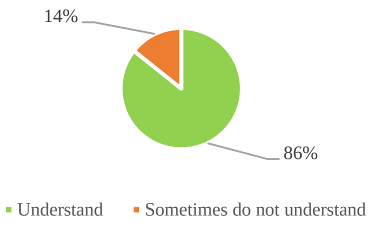

Fig. 4. Level Understanding Each RS.

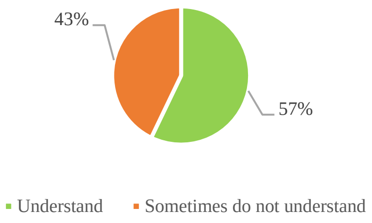

Fig. 5. Level Understanding Overall RS.

B. Communication Factor (RQ2)

This paper only focused on the language used in preparing RS Malaysia's public sector for the communication factor. We would investigate the most used prominent language. Fig. 6 shows 84% Malay and 16% English used to prepare the RS. Fig. 7 shows the policy to use a specific language for the RS. 78% did not have any policy, 11% used the internal memo, and 11% used the Service Circular No.1 / 2020 Empowerment of the National Language in the Public Service [29]. The Service Circular was to encourage writing in Malay for formal documents, including RS. Most respondents used Malay to prepare the RS, specifically for in-house projects. Most Malay RS were prepared for in-house projects because of communication among clients and requirement engineers, understanding of the RS, validation, and verification, and working culture. 44% of respondents used the English RS for outsourcing projects. Some of the outsourcing projects were prepared in English RS because the requirement engineers and developers are foreigners.

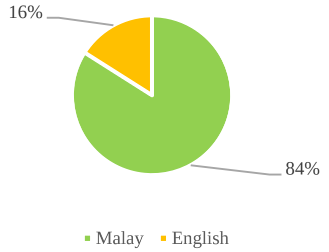

Fig. 6. Most Languages used in RS.

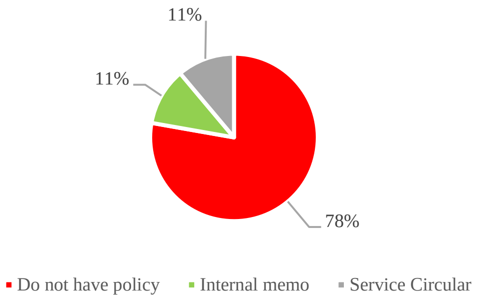

Fig. 7. Policy to use a Language for RS.

C. *Process and Procedure (RQ3)*

This study identified the methods and approaches used to prepare and verify the RS. Fig. 8 illustrates that 56% of the respondents used KRISA [30] to prepare the RS, and 44% did not have any template. Requirement engineers used only 48% of KRISA content as guidance. There were no formal instructions from the internal agencies or enforcement from the central agency to use KRISA documents. Most respondents took the initiative to implement KRISA as a handbook without any supervisor's orders. The KRISA document was used as a guide to preparing URS, BRS, SRS, and other documents. Fig. 9 shows that 89% of the respondents had verification methods for the RS. The respondents' verification methods, e.g., reviewing and confirming the RS through the committee meeting chaired by the Project Manager and Project Director. Most respondents mentioned that clients or subject matter experts verified the BRS and URS.

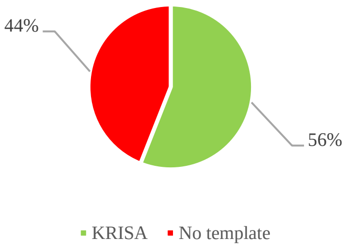

Fig. 8. Template in Preparing RS.

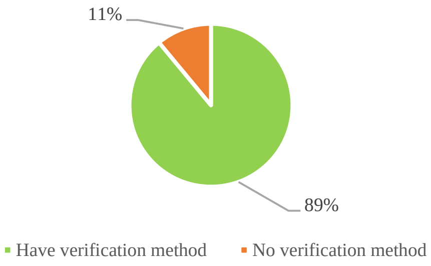

Fig. 9. Verification Method for RS.

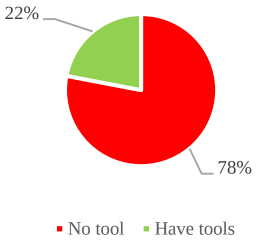

Fig. 10. Tools for RS Verification.

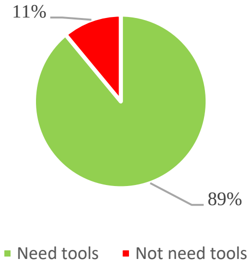

Fig. 11. Need Tools for Verifying the RS.

Meanwhile, the SRS was verified by requirement engineers. 25% of respondents found the requirement smells during verification. Fig. 10 illustrates that 78% of the respondents said no tool was used for RS verification. Verify the RS manually using Microsoft Word and Microsoft Visio tools (22%). Other tools were a mock-up interface, workflow, and flowchart. Fig. 11 depicts that 89% need tools to check RS quality, i.e., ambiguity, incomplete etc. In addition, most respondents suggested the tool could check the syntax and KRISA template-based conformance.

D. *Issues in RS (RQ4)*

This paper focused on exploring and identifying the smells in RS. Fig. 12 illustrates that 88% of the respondents (software developers) agreed that issues in RS could cause project delays or failures. Ambiguity, incomplete and incorrect were the top three familiar smells found in RS. Since ambiguity was the most significant requirement smell, we analysed the collected documents that may contain ambiguous RS. We found that 30% of the initial gathered Malay requirements were ambiguous.

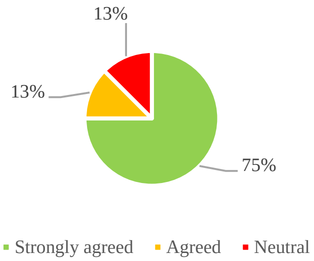

Fig. 12. Issues in RS Could Cause Project Delays or Failures.

V. Discussion

In this section, the findings are further discussed. Also, this section discusses the threats to validity.

A. *Software Developers' Experience Versus Requirement Smells*

From the perspective of software developers' experience, Fig. 13 shows that 38% of respondents with more than 10 years of experience found the ambiguity as requirement smells commonly found in RS. Fig. 13 suggests that respondents with more than 10 years of experience can identify the requirement smells in RS.

B. *Software Developers' Experience Versus Understanding the RS*

From the software developers' experience perspective, Fig. 14 illustrates that 29% of respondents with six to 10 years of experience are neutral in understanding the RS. The term neutral means sometimes understanding and sometimes not understanding the RS. The factor of neutral understanding happened caused of the inexperienced requirement engineers in developing the RS. Software developers with more than six years of experience could identify the requirement smells. Therefore, experienced software developers need further clarification from requirement engineers to understand the RS. Meanwhile, 29% of software developers with less than six years of experience understand the overall RS. Software developers with less than six years of experience typically follow the RS without screening the RS's quality.

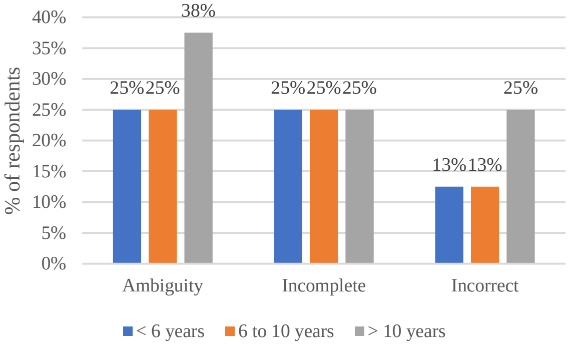

Fig. 13. Software Developers' Experience Vs Requirement Smells Commonly Found in RS.

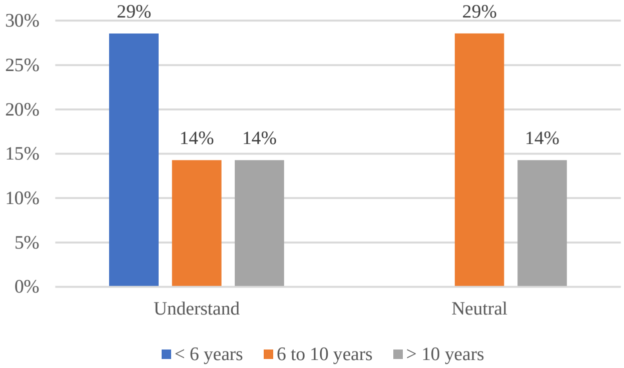

Fig. 14. Software Developers' Experience Vs Understanding the RS.

C. *KRISA Handbook*

No one instructed the requirement engineer to use any guide to prepare the RS. It is up to the requirement engineer to decide the template for creating the RS until completion. Most requirement engineers used the KRISA as a handbook to prepare the RS because it is written in Malay and easy to understand in their mother tongue. However, the KRISA handbook contains no clear or complete template for constructing the Malay RS.

D. *Threats to Validity*

During this research, there is a possibility that a few risks developed by accident due to a few circumstances brought on by various kinds of validity, such as internal, external, construct validity, and so on [31].

1) External validity: The respondents were aware of the open-ended nature of the interview questions, but they provided succinct responses nonetheless. Aside from that, respondents did not give the anticipated feedback in their responses. It may impact the low reliability of the data acquired. As a result, we questioned respondents and asked them to elaborate on their short answers.

2) Construct validity: The interview questions were vetted and tested for understandability through a preliminary interview session with some qualifying candidates before the actual interview session (illustrated in Table II).

VI. Conclusions and Future Works

A. *Conclusions*

We conducted semi-structured interviews based on RQs in the Introduction section, i.e., RQ1) Human factor, RQ2) Communication factor, RQ3) Process and Procedure, and RQ4) Issues in RS. We found that requirement smells in RS may cause a delay in the RE processes and lead to project failure. Ambiguity is the top smell in Malay RS. Based on a static analysis, 30% of the initial collected Malay RS are ambiguous. These issues can only be identified by experienced software developers, i.e., those with more than 10 years of experience. The participants mentioned that KRISA is the primary reference for preparing RS in the Malaysian public sector in terms of a template. Nevertheless, KRISA does not contain a complete requirement specification structure template compared to ISO IEEE/ISO/IEC 29148-2018, EARS, and RUPPs.

B. *Future Works*

The government needs automatic tools to detect and improve ambiguous RS based on the respondents' feedback. The respondents who wish to have tools could check the RS's quality based on the KRISA handbook. The automated tools would cover the functional and non-functional RS. Therefore, we would create the aforementioned automated tools. In addition, we are ambitious to design and prepare a Malay RS template.

Acknowledgment

The Ministry of Higher Education supported this work under the Fundamental Research Grant Scheme

(FRGS/1/2020/ICT01/UPM/02/1). We want to thank every respondent from Malaysia's public sector for their dedication and assistance.

References

[1] J. S. Sinpang, S. Sulaiman, and N. Idris, 'Detecting Ambiguity in Requirements Analysis Using Mamdani Fuzzy Inference,' Journal of Telecommunication, Electronic, and Computer Engineering, vol. 9, no. 3-4 Special Issue, pp. 157–162, 2017.

[2] E. Kamsties, "Understanding ambiguity in requirements engineering, Engineering and Managing Software Requirements," in Engineering and Managing Software Requirements. Berlin: Springer, 2005.

[3] H. Femmer, D. Méndez Fernández, S. Wagner, and S. Eder, 'Rapid quality assurance with Requirements Smells,' Journal of Systems and Software, vol. 123, pp. 190–213, 2015, DOI: 10.1016/j.jss.2016.02.047.

[4] IEEE, 'IEEE/ISO/IEC 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering', IEEE, 2018.

[5] H. Haron and A. A. Abdul Ghani, 'A Survey on Ambiguity Awareness towards Malay System Requirement Specification (SRS) among Industrial IT Practitioners', Procedia Comput Sci, vol. 72, pp. 261–268, 2015, DOI: 10.1016/j.procs.2015.12.139.

[6] A. H. A. Zahid, M. W. Haider, M. S. Farooq, A. Abid, and A. Ali, 'A critical analysis of software failure causes from project management perspectives,' VFAST Transactions on Software Engineering, vol. 6, no. 1, pp. 62–68, 2018.

[7] V. Rodríguez Montequín, J. Villanueva Balsera, S. M. Cousillas Fernández, and F. Ortega Fernández, 'Exploring Project Complexity through Project Failure Factors: Analysis of Cluster Patterns Using Self-Organising Maps', Complexity, vol. 2018, p. 9496731, 2018, DOI: 10.1155/2018/9496731.

[8] F. Ashfaq and I. S. Bajwa, 'Natural language ambiguity resolution by intelligent semantic annotation of software requirements,' Automated Software Engineering, vol. 28, no. 2, Nov. 2021, DOI: 10.1007/s10515-021-00291-0.

[9] I. Sommerville, Software engineering. Pearson Education, 2016.

[10] T. Iqbal, P. Elahidoost, and L. Lúcio, 'A Bird's Eye View on Requirements Engineering and Machine Learning, in 2018 25th Asia-Pacific Software Engineering Conference (APSEC), 2018, pp. 11–20. DOI: 10.1109/APSEC.2018.00015.

[11] C. R. Kavitha and S. M. Thomas, 'Requirement gathering for small projects using agile methods,' IJCA Special Issue on Computational Science-New Dimensions & Perspectives, NCCSE, 2011.

[12] N. Rios et al., 'Hearing the voice of software practitioners on causes, effects, and practices to deal with documentation debt,' in International Working Conference on Requirements Engineering: Foundation for Software Quality, 2020, pp. 55–70.

[13] A. Ab Aziz, 'Model Penerimagunaan Sistem Pengurusan Dokumen Dan Rekod Elektronik Dalam Sektor Awam Di Malaysia', Universiti Kebangsaan Malaysia, Malaysia, 2019.

[14] J. C. Barata, D. Lisboa, L. C. Bastos, and A. Neto, 'Agile requirements engineering practices: a survey in Brazilian software development companies,' arXiv preprint arXiv:2202.12956, 2022.

[15] V. Lenarduzzi, D. Fucci, and D. Mendéz, 'On the perceived harmfulness of requirement smell: An empirical study, in Joint 26th International Conference on Requirements Engineering: Foundation for Software Quality Workshops, Doctoral Symposium, Live Studies Track, and Poster Track, Pisa; Italy, 24 March 2020 through 27 March 2020, 2020, vol. 2584.

[16] C. Seaman and Y. Guo, 'Measuring and monitoring technical debt,' in Advances in Computers, vol. 82, Elsevier, 2011, pp. 25–46.

[17] S. Ahmad, S. A. Asmai, and N. A. Rosmadi, 'A Significant Study of Determining Software Requirements Defects: A Survey. WSEAS Press, 2015.

[18] K. Mokos and P. Katsaros, 'A survey on the formalisation of system requirements and their validation,' Science Direct, vol. 7, p. 100030, 2020, DOI: https://doi.org/10.1016/j.array.2020.100030.

[19] J. A. Aguilar, A. Zaldívar, C. T. Barba, R. Espinosa, S. Misra, and C. E. Zurita, 'A Survey About the Impact of Requirements Engineering Practice in Small-Sized Software Factories in Sinaloa, Mexico,' in ICCSA, 2018.

[20] F. Ilyas, K. Zahra, N. Ambreen, and W. H. Butt, 'A survey on current requirement process practices in software companies and requirement process problems,' in 2016 International Conference on Computational Science and Computational Intelligence (CSCI), 2016, pp. 1280–1285.

[21] K. Çamoğlu and R. Kandemir, 'A Survey of Software Requirements Engineering Practices in Turkey, 2019.

[22] A. Jarzębowicz and K. Połocka, 'Selecting requirements documentation techniques for software projects: A survey study, in 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), 2017, pp. 1189–1198. DOI: 10.15439/2017F387.

[23] U. Noorin and M. Sirshar, 'Quality Assurance in Requirement Engineering,' Global journal of computer science and technology, 2017.

[24] A. Hussain and E. O. C. Mkpojiogu, 'Requirements: Towards an understanding on why software projects fail,' in AIP Conference Proceedings, 2016, vol. 1761, no. 1, p. 020046.

[25] B. Regnell, M. Höst, J. Natt och Dag, P. Beremark, and T. Hjelm, 'Visualisation of agreement and satisfaction in distributed prioritisation of market requirements,' in Proceedings of 6th International Workshop on Requirements Engineering: Foundation for Software Quality, 2000.

[26] RapidMiner, 'RapidMiner tool.' 2006. [Online]. Available: https://rapidminer.com/.

[27] A. N. Christopher, Interpreting and using statistics in psychological research. SAGE Publications, 2016.

[28] K. Pohl and C. Rupp, Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant (1st ed.). Rocky Nook, 2011.

[29] Jabatan Perkhidmatan Awam Malaysia, 'Pekeliling Perkhidmatan Awam Bilangan 1 Tahun 2020: Pemerkasaan Penggunaan Bahasa Kebangsaan Dalam Perkhidmatan Awam'. 2020.

[30] Malaysian Administrative Modernisation and Management Planning Unit (MAMPU), Public Sector Application System Engineering Handbook (KRISA) - 2nd Chapter: Business Requirements Specification. MAMPU, 2019.

[31] K. Petersen and C. Gencel, 'Worldviews, research methods, and their relationship to validity in empirical software engineering research, in 2013 joint conference of the 23rd international workshop on software measurement and the 8th international conference on software process and product measurement, 2013, pp. 81–89.