



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

A Systematic Literature Review on the Scalability Issues in Software Requirements Prioritization

Master's thesis in Computer Science and Engineering

NAYEM NURUL KADER

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

MASTER'S THESIS 2022

A Systematic Literature Review on the Scalability Issues in Software Requirements Prioritization

NAYEM NURUL KADER



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

A Systematic Literature Review on the Scalability Issues in Software Requirements
Prioritization
NAYEM NURUL KADER

© NAYEM NURUL KADER, 2022.

Supervisor: Fadhl Hujainah, Department of Computer Science and Engineering, and
Volvo Car Corporation, Sweden
Examiner: Gregory Gay, Department of Computer Science and Engineering

Master's Thesis 2022
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in L^AT_EX
Gothenburg, Sweden 2022

A Systematic Literature Review on the Scalability Issues in Software Requirements Prioritization.

NAYEM NURUL KADER

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Context: Prioritizing requirements is an important process that plays a vital role in producing a successful quality system by selecting high-risk requirements for implementation. Several studies have been conducted to analyze the performance of existing RP techniques. The inability to address a large number of requirements (Scalability issue) has been revealed as one of the major problems facing most of the existing requirement prioritization (RP) techniques.

Objective: The scalability is a great challenge that has led to the RP process becoming inapplicable and inefficient in the industry, as most industrial projects nowadays have hundreds of requirements to be implemented. This paper aims to critically analyze the scalability issue in RP in terms of a) revealing and analyzing the factors that induce to the scalability issue while prioritizing a large number of requirements along with revealing the impact of each identified factor in raising the scalability issue, b) identifying and analyzing the existing techniques proposed to handle the scalability issue with respect to their usage contexts, the features used and their consequence in handling the scalability issue in their RP processes, c) investigating the evaluation extent of the identified techniques in terms of being evaluated and implemented in the industry settings, d) assessing the capabilities of the the identified RP techniques in considering the listed causative factors of the scalability issue in their performance evaluation, and e) revealing the limitations and future recommendations for further research.

Method: A review protocol was designed based on the standard review guidelines by Kitchenham. Initially, 258 potential related studies were compiled through a step wise searching process. Thereafter, by thorough selection procedure, 22 relevant studies were finally selected to address the listed research questions.

Results: The findings revealed four causative factors of scalability issue in RP with their descriptions: number of comparisons, time, human efforts required, and scarcity of the automation, along with revealing the impact of each identified factor in provoking the scalability issue. We found that the number of comparisons is the most significant factor that can lead to the inability to deal with large-scale requirements in prioritization with an impact degree of 48% percentage in leading to the scalability issue. Twenty three techniques that focused on addressing the scalability issue have been identified and analyzed critically in terms of their software development usage contexts, features used and the usage consequences of the

utilized features in addressing the scalability issue, evaluation extents, and capability in addressing the identified causative factors of the scalability issue during their evaluation performance, and limitations. It has been found that most of the identified techniques were evaluated within the industrial settings with a percentage of 82%. However, the results reveal that these techniques still face serious limitations with respect to rank update, requirement interdependencies, substantial dependence on expert engagement, and error-prone and sensitivity to judgemental errors of the existing techniques.

Conclusion: The scalability issue in the RP process has been elaborated on precisely in this study. Based on the findings, most of the identified techniques have not sufficiently covered the causative factors in their performance evaluations as at most up to 2 factors have been considered (mostly attributed to the fact that they are primarily considered as proofs-of-concept). Thus, a comprehensive performance evaluation of these techniques is needed to ensure their capabilities in addressing the scalability issue sufficiently in terms of considering the four identified causative factors. Additionally, adoption of the present strengths that have been found and attention to their underlying limitations are highly necessary. Thus, the current work represents several suggestions that can act as guides for the advancement of RP techniques in addressing the identified limitations.

Keywords: Scalability, requirement prioritization (RP), RP techniques, limitation, usage consequence, usage context.

Acknowledgements

From the beginning to the end of this master thesis it has been a really long and faithful journey. First of all, I would like to thank my supervisor, Fadhl Mohammad Omar Hujainah, for giving me the opportunity to do my master's thesis under his supervision. Throughout this period, I have received really encouraging and unlimited support from my supervisor. This master thesis could not have been completed without his cooperation.

I am deeply grateful to my parents for their support, appreciation, encouragement, and keen interest in my academic achievements. Finally, I really thank my wife for her moral support throughout the master thesis time being.

NAYEM NURUL KADER, Gothenburg, September 2022

Contents

List of Figures	xi
List of Tables	xii
1 Introduction	1
1.1 Context	1
1.2 Problem Statement	2
1.3 Purpose of the Study	3
1.4 Contribution	3
1.5 Structure of the Thesis	4
2 Background and Related Work	5
2.1 Requirement Prioritization	5
2.2 Related Works	7
3 Research Questions	11
4 Method	13
4.1 Identification of the need of the SLR and research question formula- tion	13
4.2 Search Strategy	14
4.3 Study Selection	15
4.3.1 Inclusion and Exclusion Criteria	15
4.3.2 Quality Assessment Criteria	17
4.4 Data Extraction and Synthesis	19
5 Results	20
5.1 Overview of the selected studies	20
5.2 RQ1. What are the causative factors of the scalability issue in RP and what is the meaning of each factor?	21
5.3 RQ2. How do the identified factors lead to the scalability issue and which factors have the highest impact?	22
5.4 RQ3: What are the techniques that have been proposed to address the scalability issue in the RP process and what are their software development usage contexts?	25

5.5	RQ4: What are the features (e.g., type of the adapted method, algorithm) used by the identified RP techniques in RQ3 to handle the scalability issue and what are their usage consequences in solving the scalability issue?	29
5.6	RQ5. Which of the RP technique from RQ3 have been evaluated in terms of the scalability issue, and to what extent these techniques have covered the identified causative factors of scalability issue in their evaluation processes?	37
5.7	RQ6. What are the limitations of the identified techniques in RQ3? .	46
6	Threats to validity	50
7	Conclusion	52
7.1	Discussion	52
7.2	Future Recommendations	53
7.3	Conclusion	54
	References	56
A	Appendix 1	I

List of Figures

1.1	Statistics of the existence of the Scalability Issues in PR [6].	3
4.1	Review Protocol.	13
4.2	Search Strategy.	15
4.3	Study Selection Process.	16
5.1	Publication Channels.	20
5.2	Publication years of selected primary studies.	21
5.3	Importance degree of Scalability Causative Factors.	24
5.4	Percentage of Used Contexts in the Selected Studies.	29
5.5	Categories of the RP techniques based on the features used in addressing the Scalability issue	30
5.6	Percentage of the evaluation methods in evaluating the existing technique.	37
5.7	Evaluation methods in evaluating the existing technique.	38
5.8	The percentage of the evaluation environment settings used in assessing the performance of the existing techniques.	39
5.9	Percentage of the causative factors being catered in addressing the scalability issue in all identified techniques.	40
5.10	Number of the causative factors being covered in addressing the scalability issue by each identified technique.	41

List of Tables

2.1	Analysis of existing review studies in terms of their focus and findings.	9
2.1	Analysis of existing review studies in terms of their focus and findings (continued).	10
3.1	Research questions and their motivations.	11
3.1	Research questions and their motivations (continued).	12
4.1	Inclusion and Exclusion Criteria.	17
4.2	Quality checklist questions.	17
4.3	Quality Points of Selected Studies.	18
4.3	Quality Points of Selected Studies (continued).	19
5.1	Factors that considered for the scalability issues.	22
5.2	Analysis of Identified techniques in terms of the features used and their usage consequences.	33
5.2	Analysis of Identified techniques in terms of the features used and their usage consequences (continued).	34
5.2	Analysis of Identified techniques in terms of the features used and their usage consequences (continued).	35
5.2	Analysis of Identified techniques in terms of the features used and their usage consequences (continued).	36
5.3	Analysis of evaluation extent for each identified techniques.	42
5.3	Analysis of evaluation extent for each identified techniques (continued).	43
5.3	Analysis of evaluation extent for each identified techniques (continued).	44
5.3	Analysis of evaluation extent for each identified techniques (continued).	45
5.4	Analysis of the existing limitations of each identified selected studies.	47
5.4	Analysis of the existing limitations of each identified selected studies (continued).	48
5.4	Analysis of the existing limitations of each identified selected studies (continued).	49

1

Introduction

1.1 Context

Requirements Prioritization is one of the most important phases in software system development that deal with the process of managing the systems. Capturing, analyzing, and prioritizing the system's requirements are the main factors that lead to making the system acceptable to the stakeholders [1], [2]. Requirements prioritization (RP) is an important process that assists in developing a good quality system by selecting high priority requirements to be implemented in the earlier releases on the basis of the stakeholders [3]–[5].

RP also helps overcome the challenge of creating a successful system that contains a large number of requirements by using the available resources to implement the identified most important and risky requirements first, rather than wasting resources on other less important requirements [6]–[8]. A number of different techniques have been proposed to prioritize the requirements efficiently and accurately. Several studies have been conducted to analyze the performance of the existing RP techniques. Scalability issue has been revealed as one of the most serious challenges faced by the existing RP techniques. However, none of the existing studies have emphasized analyzing the issue of scalability in prioritizing the requirements. Scalability refers to the inability to deal with a large number of requirements. The sets of requirements in RP are divided into three sets [1], [6], [9]:

- Small set of requirements: the number of requirements define the following range: (number of requirements < 15)
- Medium set of requirements: the number of requirements is defined between the following range: ($15 \leq$ number of requirements < 50)
- A large set of requirements: the number of requirements is defined between the following range: (number of requirements ≥ 50)

To prioritize the requirements, the stakeholders of the system must rate the importance degree of each requirement and perform the pairwise comparisons. This process becomes complex as the number of requirements increases. Although most software products have become more complex (e.g., containing a large set of requirements), most existing RP techniques can only work well with a small number of requirements, such as the analytic hierarchy process (AHP) technique [1], [6]. The scalability issue has a significant impact on the implication of the prioritization process in industrial projects, as most current projects include a large number of

requirements to be prioritized [1], [6]. Although most of the software products became more complex in terms of containing more than one hundred requirements which are considered a large set of requirements in RP, most of the RP existing techniques can only work well with a small number of requirements [1], [6], [10]. To improve the current RP techniques and come out with an efficient solution for the scalability issue, there is indeed a need to identify and analyze the factors that lead to raising this issue in the requirements prioritization. Thus, this research is aimed at providing a comprehensive investigation on the scalability issue in the form of the systematic literature review (SLR) to critically analyze the reason that leads the existing techniques to be not able to work with a large set of requirements by identifying and analyzing the factors that cause this problem. The result of this research will help the researchers in the industry and academic sectors by providing a critical investigation on the scalability issue to improve the current state of RP practices and art.

1.2 Problem Statement

Prioritizing the requirements is a critical process that assists in developing a good quality system by selecting the high-risk requirements to be implemented. Different techniques have been proposed to conduct the RP process. The inability of dealing with a large number of requirements (scalability) is currently one of the main issues in most existing RP techniques. Scalability issue is a tremendous challenge that leads the prioritization process to be impracticable on the industrial side since most industrial projects nowadays contain hundreds of requirements to be implemented. Recently, two research studies conducted by Bukhsh et al [1], Hujainah et al [6], revealed the percentage of the poor handling of scalability issues. We can see in (Figure 1.1) that on the basis of the 107 existing RP techniques, the percentage of not being able to address the scalability issue is 93%, and the percentage of successfully handled the scalability is only 7% as stated. Thus, the scalability issue is a serious challenge faced by the existing RP techniques when they deal with large-scale requirements.

Various review studies (e.g., [1], [6]) have been conducted to investigate the performance of the existing RP techniques and analyze their strengths and limitations. At glance, these studies provide a comprehensive analysis of the RP techniques in terms of their limitations generally. However, to date, there has been no systematic literature review (SLR) conducted specifically concerning investigating the scalability issue in RP as shown in Table 2.1 that presents a summary of findings of the existing review studies. Thus, this research aims to fill this gap by identifying and investigating the factors that cause the scalability issue in the prioritization process and will specifically elaborate on the impact of each identified factor in raising the scalability issue while prioritizing a large number of requirements. In addition, our study will analyze the existing techniques that have been proposed to handle the scalability issue with respect to identifying the features used by the techniques in solving the scalability issue and revealing their usage consequences. Also, this research will reveal the limitations of the RP techniques introduced to address the

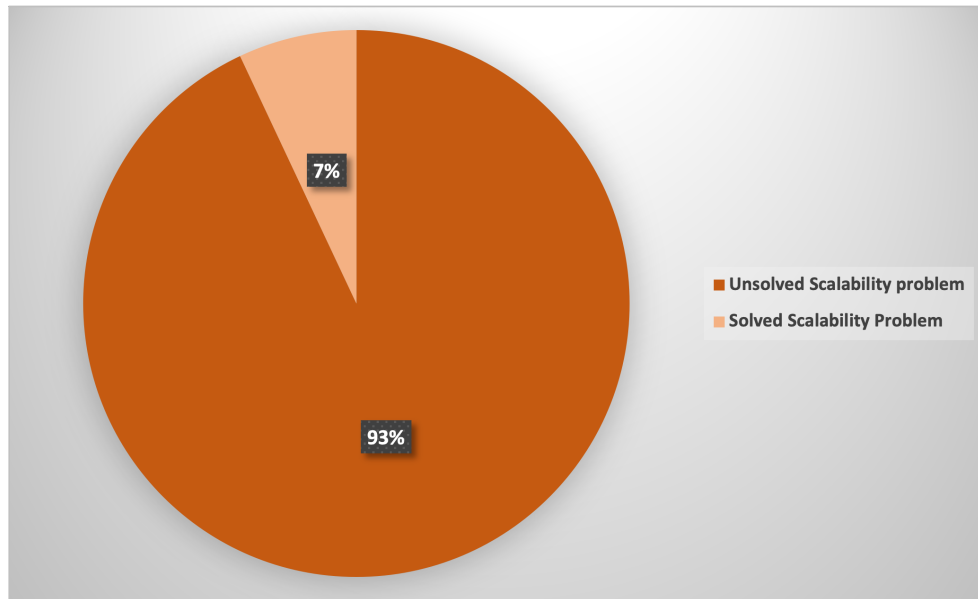


Figure 1.1: Statistics of the existence of the Scalability Issues in PR [6].

scalability issue.

1.3 Purpose of the Study

The purpose of this study is to provide a comprehensive investigation of the scalability issue in the requirements prioritization field. The goal of this thesis can be summarized as follows:

- Investigating the scalability issue in RP in terms of revealing and analyzing the factors that cause the scalability problem.
- Analyzing the existing techniques that have been proposed to handle the scalability issue in terms of identifying the features used by the techniques in solving the scalability issue and revealing their usage consequences.
- Assessing to what extent the identified RP techniques (that proposed with aim of handling the scalability issue) have been validated in addressing the scalability issue and considering the causative factors of the scalability issue in their prioritization processes.
- Revealing the limitations of the RP techniques that were introduced to address the scalability issue and providing a future recommendations for further research to be conducted.

1.4 Contribution

This research will help the researchers in the industry and academic sectors by providing a critical investigation on the scalability issue to improve the current state of RP practices and art. Such investigation will assist in providing a precise investigation of the scalability difficulty in RP in terms of uncovering and assessing

the components that contribute to the problem. Also, this study adds new value in terms of assessing the existing techniques proposed to address the scalability problem in terms of identifying the features employed by the techniques in resolving the scalability problem and disclosing the implications of their application. Additionally, this SLR can assist the researcher and practitioners in determining whether the identified RP techniques (offered to address the scalability issue) have been validated in addressing the scalability issue and taking into account the scalability issue's causative components in their priority procedures and outlining the drawbacks of the RP techniques that were adopted to address the scalability issue, as well as making recommendations for future research.

1.5 Structure of the Thesis

This thesis is structured as follows: section 1 illustrates the introduction of this systematic literature review (SLR). Section 2 presents background and related work. Section 3 describes the research questions. Section 4 has discussed about the method used in conducting this SLR. Section 5 reflects the results of this SLR with answering the research questions. Section 6 gives a conclusion of this study.

2

Background and Related Work

This chapter provides background and related works of this research, starting with providing an overview of the requirements prioritization (RP) in software development, and the existing common RP techniques. Thereafter, a critical investigation of the existing review studies on RP is presented to discuss the need for conducting this SLR.

2.1 Requirement Prioritization

Prioritizing the requirements is an important stage which helps to make the decisions when developing a system. Prioritizing requirements is generally considered as a difficult and complex decision-making process [11]. An important requirement in one release for one customer may not be as important in the following release for another customer [11]. Sometimes, customers may get distrustful regarding the prioritization because they are concerned that only the most important criteria will be implemented on the other hand developers become skeptical because they do not want to reveal that they are unable to implement all of the criteria [11]. There is a growing need for systems capable of prioritizing the requirements in the development of commercial software systems. Efficient and reliable techniques for prioritizing requirements are strongly demanded by practitioners [12]. Prioritizing the requirements is a critical process that assists in developing a good quality system within limited resources (such limited time, budget, and staff) by selecting the high-risk requirements to be implemented. When the project's budgets and time run out, there is a need to address the most important requirements through the requirement prioritization to satisfy the consumers.

Requirement prioritization aspects: Different prioritization aspects are considered while conducting the prioritization process. Prioritizing the requirements can be performed by employing two types of aspects: technical aspects and commercial aspects [13]. Technical aspects concern about the time, cost value, effort, resources, decision making. Another aspect [13], [14]. On the other hand, commercial aspects is concerned about customer's satisfactions, financial benefits and marketing [13], [14]. Balancing between these two types of aspect is needed to be captured in formulation the prioritization process.

Numerical assignment technique: The numerical assignment technique is one of the widely used technique. This technique divides the requirements into differ-

ent priority groups before conducting the prioritization process: the critical group, the standard group, and the optional group. Then, the involved stakeholders will be asked to rank the requirements by grouping them into the defined three priority groups [12], [14]. Although this technique is simple, it has a number of drawbacks. For example, the stakeholders think that most of requirements are considered to be significant, so they will categorize most of the requirements to the critical priority group [12], [14]. Hence, the implementation of all requirements will be significant, so this will affect the efficiency of this technique in term of identifying the most important requirements. As well as, this technique does not provide priority value for each requirement but all requirements which are in each group have the same priority, since the priority value is given for group priority not for each requirement [12], [14].

Analytic Hierarchy Process (AHP): AHP is the most common technique in RP domain [11]. This technique is based on the pairwise comparisons. To perform the pairwise comparisons, AHP first identifies the attributes and alternatives for each aspect and uses those to form a hierarchy. Thereafter, users choose their preference for each set of attributes by assigning a judgment scale, often 1 to 9, where 1 represents an equal value and 9 represents the highest value [12]. The required number of pair wise comparisons are determined by this formula $(n*(n-1)/2)$, where n is indicated to the number requirements [12]. Following that, AHP interprets the user's evaluations to numerical values and assigns a numerical priority to each item of the hierarchy. AHP is used to analyze the requirements and determine which are the most important and to what extent and mostly used in the small set of requirements [12]. This technique faces the scalability issue as the number of comparisons increase dramatically with large number of requirements that needs to be prioritized and this make technique to consume much time and efforts [12] ref.

Hierarchy Analytic Hierarchy Process (AHP): Hierarchy AHP technique is introduced by Karlsson et al [12]. The purpose of this techniques is to overcome the scalability issues of AHP techniques. In AHP, when the number of requirements increase then the number of comparison increases as well which will affect the efficiency of AHP technique. Therefore, this technique reduces the number of the comparisons by prioritize the requirements that are only in the same hierarchy level based on the accumulated scores of each requirement across relevant stakeholders [11]–[13].

Minimal Spanning Tree: The Minimal spanning tree technique introduced by Karlsson et al [12]. This technique is to eliminate the redundancy of pair wise comparisons of AHP technique. This redundancy affects the AHP technique by causing scalability issue as the number of pairwise comparisons increases. Minimal spanning tree technique solves the redundancy issue without performing all redundant comparisons. This will help to reduce the number of comparisons to $(n-1)$ which is much less compared to AHP technique [11]–[13]. Hence, minimal spanning tree is one of the fastest techniques to prioritize the requirements and it is work well with large number of requirements [11]–[13]. However, AHP technique provides

better reliable than this technique. Also, it is more sensitive to judgmental errors due to inconsistencies in rank result [12].

Cumulative Voting: Another requirements prioritization technique is commutative voting which is known as hundred-dollar technique as well [14]. This technique (100-dollar) is considered as a simplest technique for determining the requirements. The basic concept for this technique is that stockholders asked to assume their unit to distribute to the requirements (e.g., 100 dollars, 1000 points) [13], [14]. This point is referred to as a priority of the requirements, and this outcome is presented using a ratio scale by which it can be shown how much in one requirement [13], [14]. This technique is straightforward and suitable to be used with small set of requirements, but it is not advisable to be used with large set of requirement [13], [14].

2.2 Related Works

Five review studies were conducted recently in the RP field.

Nidhra et al [4] presented a survey and critical evaluation in RP, which critically evaluated the existing studies in the context of multiple disciplines, for instance, software engineering, product manufacturing, and engineering. The aim of this study was to explore the strengths of the existing techniques in the context of RP. Through this study, 158 RP existing techniques were evaluated: 67 techniques were with the focus of multiple studies. Moreover, findings revealed that the AHP, cumulative voting, QFD, numerical assignment, and planning game techniques are mostly used by the researchers in the contexts of RP techniques and revealed that scalability is considered one of the most serious challenges for most of the existing techniques.

Bukhsh et al [1] presented a systematic literature review on RP techniques and evaluated them empirically. This SLR focused only identifying and evaluating the RP techniques between 2007 to 2019. The performance of the selected techniques was evaluated in terms of ease of use, scalability, and effectiveness. This study concluded that the AHP technique is one of the the most accurate and extensively used technique for RP in the industry and highlighted scalability as the major issue when dealing with a large set of requirements.

Achimugu et al [10] also presented SLR which is considered a fundamental SLR on RP. This SLR investigated and analyzed the existing RP techniques in terms of limitations, taxonomies, and processes. The results demonstrated the steps of the RP process. Forty-nine techniques have been identified and analyzed in terms of the limitations and findings indicated that scalability, inability of dealing with rank updates during requirements evolution, coordination among stakeholders and requirements dependency issues are main limitations in RP.

Hujainah et al [15] conducted an SLR study on revealing the impact of the RP in software development, type of the stakeholders involved during the prioritization process, existing techniques, and challenges. In this SLR, 122 papers was selected and 108 RP existing techniques was revealed and analyzed in terms of the limitations and prioritization aspects. This SLR's findings stated that the existing techniques are still facing serious issue related to scalability issues and suggest for critical investigation to be conducted to reveal causative factor of scalability issue.

Complementing the existing works, Hujainah, Bakar, and Abdulgabber conducted a study [16] with aim of analyzing the existence and execution process of handling the requirements interdependencies (RI) in existing RP techniques. The analysis has been conducted in 65 existing techniques. The results presented that there are only four(4) techniques that handle this RI in RP. However, it has been reported of inability of these techniques in dealing with large set of requirements (scalability issue).

Table 2.1 reflects the analysis of the five existing review studies in terms of their focus, covered findings, and similar and different findings of this SLR and existing review studies. As can be noticed from the analysis of the existing review studies (depicted in Table 2.1), the main aim of those related studies was to identify and assess the performance of the RP techniques in general without providing a precise investigation of the scalability issue. This SLR aim is to consider the scalability issue specifically in terms of revealing the causative factors of the scalability issue and analyzing the existing techniques that have been proposed to address the scalability issue with respect to identifying the features used by the techniques in solving the scalability issue and revealing their usage consequences. Also, this SLR aims to present a detailed analysis in assessing to what extent the identified RP techniques (that were proposed with the aim of handling the scalability issue) have been validated in addressing the scalability issue and considering the causative factors of the scalability issue in their prioritization processes.

Table 2.1: Analysis of existing review studies in terms of their focus and findings.

Study Reference	Focus	Covered Findings	Similar Findings to those of this SLR	Uncovered Findings Added into this SLR
Hujainah et al [16]	Analyzing the Requirement Interdependencies in RP Techniques.	Highlighted detailed overview on four selected techniques that handled requirement interdependencies in RP	Overview on RP	-Identification of the factors that cause the scalability problem. -A detailed analysis of the impact of each identified factor in raising the scalability issue. -Detailed analysis of the existing techniques that have been proposed to identifying the features used by the techniques in solving the scalability issue and revealing their usage consequences.
Bukhsh et al [1]	Analyzing RP techniques in the period of Sept 2007–June 2019.	-Empirical evidence has been conducted in the literature that was published after 2007 regarding RP techniques. -A detailed analysis of the evaluation approaches used in assessing the RP techniques.	Overview of RP and scalability issue by only reporting the scalability as one of the major issues.	-Assessing to what extent the identified RP techniques have been validated and considering the causative factors of the scalability issue in their prioritization processes.
Nidhra et al [4]	Comprehension of the full range of RP techniques that are available across multiple domains of product manufacturing, and engineering, software engineering	-A detailed analysis of the evaluation measures of the RP techniques and their performance outcomes. -A detailed exploration of RP techniques used in multiple domains: including product manufacturing, and engineering, software engineering.		

Table 2.1: Analysis of existing review studies in terms of their focus and findings (continued).

Study Reference	Focus	Covered Findings	Similar Findings to those of this SLR	Uncovered Findings Added into this SLR
Hujainah et al [15]	Analyzing the characteristics of participating stakeholders in RP, the significance of RP in the development process, RP techniques, and their challenges.	<ul style="list-style-type: none"> -Details analysis of the RP importance of the RP. -The limitations and benefits of the identified techniques. -Analyzing the participating stockholders in RP. -Detailed analysis of the usage context of RP techniques and selected studies. 		-A detailed analysis of the limitations of the RP techniques that were proposed to handle the scalability issue.
Philip Achimugu et al [10]	Identifying and analyzing the performance of existing RP techniques.	<ul style="list-style-type: none"> -Analysis of 49 techniques with respect to the descriptions and limitations of the identified techniques and the steps of the RP process. 		

3

Research Questions

This Chapter describes the research questions of this research. Six research questions (RQ) were formulated to accomplish the defined goal of this study. Table 3.1 presents the formulated research questions associated with their motivations.

Table 3.1: Research questions and their motivations.

Research Questions	Motivation
RQ1. What are the causative factors of the scalability issue in RP and what is the meaning of each factor?	To investigate and identify the factors that cause the scalability issue in the prioritization process.
RQ2. How do the identified factors lead to the scalability issue and which factors have the highest impact?	To specifically elaborate on the impact of each identified factor in leading to the scalability issue while prioritizing a large number of requirements.
RQ3. What are the techniques that have been proposed to address the scalability issue in the RP process and what are their software development usage contexts?	To identify the existing techniques that have been proposed to handle the scalability issue with revealing their usage contexts. Usage context is refers to the software development context that technique has been proposed for or the contexts where the technique has been implemented in.
RQ4. What are the features (e.g., type of the adapted method, algorithm) used by the identified RP techniques in RQ3 to handle the scalability issue and what are their usage consequences in solving the scalability issue?	To identify the features used by the techniques in solving the scalability issue and revealing their usage consequences in addressing scalability issue. Usage consequence here refers to the effect of using the feature in making the technique able to handle the scalability.

Table 3.1: Research questions and their motivations (continued).

Research Questions	Motivation
RQ5. Which of the RP technique from RQ3 have been evaluated in terms of the scalability issue, and to what extent these techniques have covered the identified causative factors of scalability issue in their evaluation processes?	To investigate evaluation extent of the identified techniques from the RQ3 in terms of being evaluated or not. Additionally, exposing evaluation environment settings in terms of revealing the name of the used projects/data sets in evaluation, and types of the evaluation environment settings with respect to academic settings (the source of the used projects/data sets are from the academic, institution or contexts) and industrial settings (source of the used projects/data sets in evaluation from the industry organization or contexts). Also, to assess to what the extent these techniques have covered the listed causative factors of the scalability issue from RQ1.
RQ6. What are the limitations of the identified techniques in RQ3?	To reveal the limitations of the RP techniques introduced to address the scalability issue.

By answering those above-mentioned research questions, we aim to identify the scalability issues in the requirements prioritization field so that we tried to investigate and identify the factors that cause the scalability issue in the prioritization process. We intend also to specifically elaborate on the impact of each identified factor in leading to the scalability issue while prioritizing a large number of requirements. Moreover, we aim to identify and analyze the existing techniques that have been proposed to handle the scalability issue in terms of revealing features (process or method) used by these techniques in solving the scalability issue and revealing usage consequences of the used features in handling the scalability issue. Furthermore, we aim to analysis the evaluation’s extent of identified techniques in terms of being evaluated in addressing the scalability efficiently and assess their capabilities in covering the causative factors of the scalability issue in their evaluation performance. In addition, our goal is to reveal the limitations of the RP techniques introduced in addressing the scalability issue along with recommending future recommendations to encourage the research in improving the performance of the technique in addressing the scalability efficiently.

4

Method

This chapter illustrates the methodology used for this research. For the review process for this study, we used to choose SLR as the research method. Figure 4.1 depicts the designed review protocol used to conduct this research. The designed review protocol was based on the guidelines of SLR proposed by Kitchenham [17]. This protocol has four main components: research questions, search process, study selection and procedures, data extraction and synthesis, and dissemination of the results. Each of those components has specific sub-components, which provide a wide explanation of each of those components as well as sub-components.

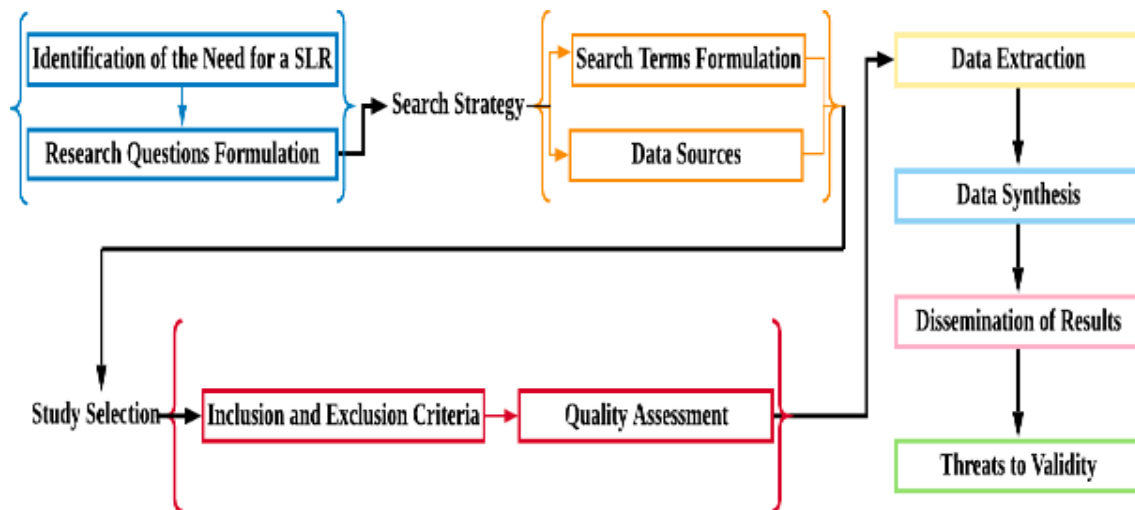


Figure 4.1: Review Protocol.

4.1 Identification of the need of the SLR and research question formulation

The motivation and the need for conducting this SLR were elaborated in the previous section (related work), in which the previous review studies were analyzed with respect to their focus and findings. The analysis revealed that recent SLRs have not specifically emphasized the scalability issue of RP. This SLR emphasizes this gap by investigating the scalability issue in the form of the SLR to critically analyze the reason that leads the existing techniques to be not able to work with a large set of requirements by identifying and analyzing the factors that cause this

problem. The research questions of this SLR were formulated on the basis of the stated objective.

4.2 Search Strategy

For this review study, the search process was conducted on the basis of two sub-components, namely, the search terms to be used for the search and the specific literature sources to be searched. The major challenge is to conduct a text search to provide accurate search terms to ensure the quality of the extracted studies. To overcome this challenge, we formatted the search terms in a step wise manner on the basis of listed research questions and standards, which include the following steps:

- Specification of prime terms based on the listed research questions.
- Identification of alternative or equivalent spelling and synonyms of each prime term.
- Verification of the terms in relevant studies.
- Utilization of the Boolean OR/AND operators to incorporate and link the terms.
- Discussions among authors to specify a stable final search term.

The final list of the search terms formulated is as follows: "Requirements Prioritization" OR "Requirements selection" OR "Scalability of requirements prioritization" AND ("issue" OR "challenges" OR "Limitations" OR "shortcomings") OR "scalable requirements prioritization" AND ("techniques" OR "methods" OR "frameworks" OR "models" OR "attributes" OR "factors" OR "features").

We collect the potential related existing works in two stages as depicted in Figure 4.2: 1) considering final selected studies of the most related SRL study and 2) conducting automatic searches.

In search stage 1, we did not conduct searching from scratch as we have considered the final selected studies that represent the searching and filtering results of the existing SLR conducted by Hujainah et al [15], in which the searching process was conducted based on generic search terms for requirements prioritization from 1984 to 2018, and 112 studies were selected as final primary studies.

In the second searching stage (search string 2), we have conducted automatic searches for extracting the most recent existing works from 2018 to 2021 with the use of the above-formulated search terms. A certain number of literature sources is selected to be searched, and among those sources, we chose four literature sources: IEEE Xplore, Scopus, ACM digital libraries, and ScienceDirect. These selected literature sources are considered to be the most relevant digital libraries or resources for software engineering research. For the searching process considered various kinds of publications: published journal papers, conference papers, workshops, and book chapter, and IEEE Bulletins. We executed the searching by applying the formulated the search terms to the titles, keywords and abstracts of the studies in the selected

literature databases and then retrieved studies that contains at least one of the formulated search terms.

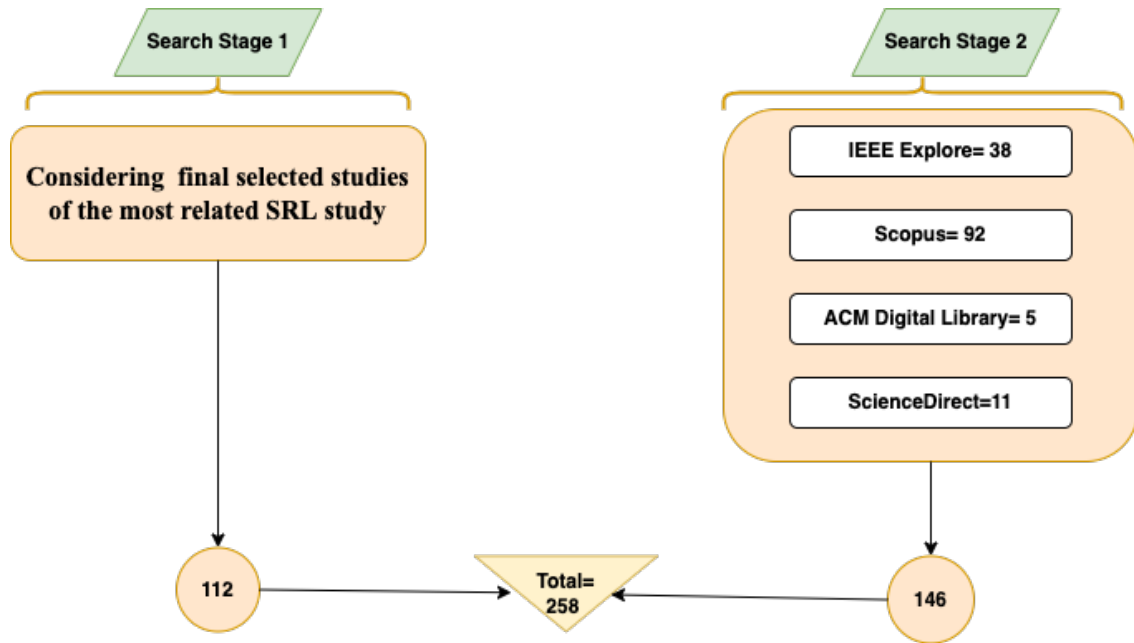


Figure 4.2: Search Strategy.

4.3 Study Selection

The purpose of this phase is to scrutinize the studies compiled from the literature sources. Figure 4.3 presents the selection procedure used in this research to reveal the most relevant studies to the specified domain of this SLR. The selection procedure is executed on the basis of two sub-activities: inclusion and exclusion criteria and quality assessment criteria.

4.3.1 Inclusion and Exclusion Criteria

With the execution of the search strategy phase, 258 potential related studies have been compiled. Table 4.1 depicts the roles of the inclusions and exclusion criteria, which have been formulated on the basis of the listed research questions and specified research domain and following the guidelines by Kitchenham [17]. The inclusion and exclusion criteria were implemented in this stage to access and analyze the collected potential studies. The inclusion criteria and exclusion criteria were formed on the basis of the research question and research domain as well. The defined inclusion and exclusion criteria were applied to all studies compiled from the selected sources. The studies that satisfied the defined inclusion criteria were included. The investigation of the studies was conducted by precisely reading the title, keyword, abstract, and conclusion of each retrieved study. We have included the English published studies and excluded the studies whose texts are unavailable and in languages other than English. On the other hand, the studies that concern the RP

process without presenting support, investigation, or discussing the scalability issue were excluded. In contrast the studies that mainly focus on the scalability issue of the RP were included, along with the empirical studies that evaluate the scalability performance of the existing RP techniques on the basis of carefully screening the title, keyword, abstract, and conclusion of each collected study.

Additionally, the studies that are able to address at least one of the specified research questions were included and excluded those studies that were not relevant to the research questions. Also, the studies in the grey literature (i.e., published website works, non-peer-reviewed works, in progress working studies, and studies that lack full bibliographic details such as volume, issue numbers, and publication date/type) were excluded. Concerning the duplicate studies, the duplicate check took place, and the most complete and recent version of the study was only included and excluded other multiple versions. After the application of the defined inclusion and exclusion criteria, 47 studies were selected. Thereafter, we have checked the references of each selected study to ensure all potential related studies have collected, even the studies that may have been overlooked during the initial round of the search process. As a result, only four relevant studies were identified, making the overall number of the remaining relevant studies to be 51, which is going to be analyzed critically in next filtering step 4.3.2.

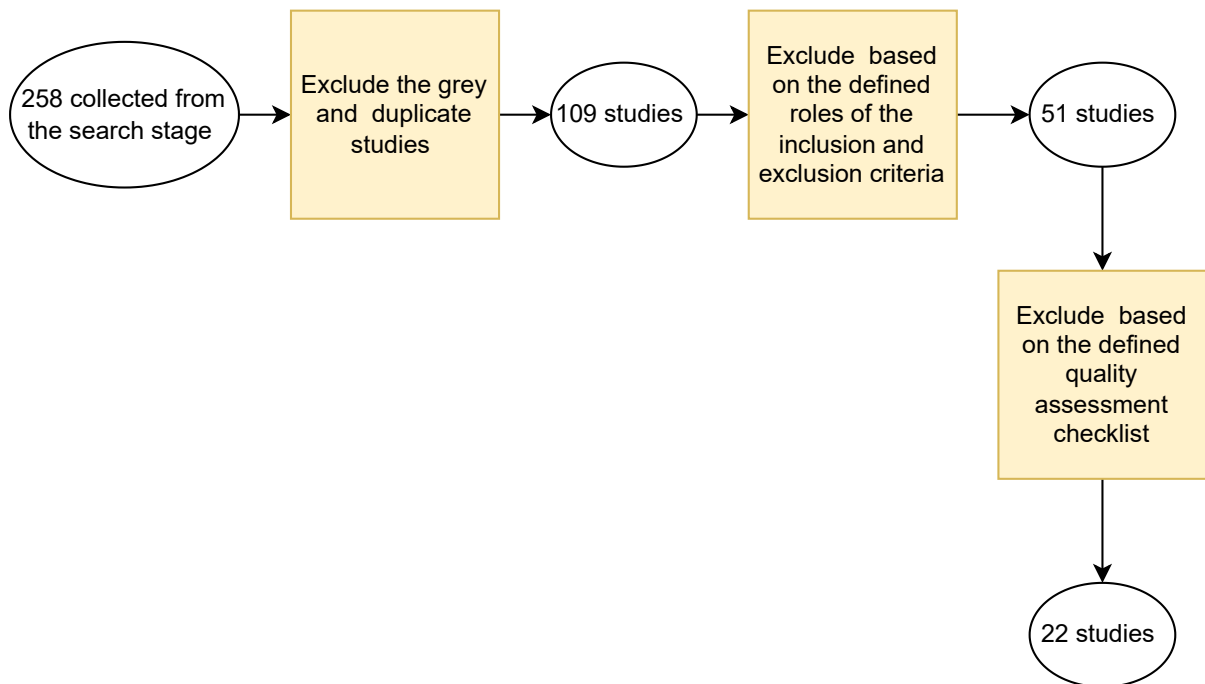


Figure 4.3: Study Selection Process.

Table 4.1: Inclusion and Exclusion Criteria.

Inclusion Criteria (IC)
IC 1. Studies with available full text.
IC 2. Studies that are written and published in English.
IC 3. Studies that concern the RP with presenting further details on the scalability issue.
Exclusion Criteria (EC)
EC 1. Studies whose full texts are unavailable.
EC 2. Studies that are written and published in non-English.
EC 3. Studies that focus on the RP process without providing further details on scalability issues.
EC 4. Duplicate Studies.
EC 5. Grey Studies.

4.3.2 Quality Assessment Criteria

In this SLR, the chosen studies from inclusion and exclusion criteria (51 studies) are going to be evaluated based on a quality checklist, which was formulated on the basis of the quality assessment guidelines in Kitchenham [17] and the listed research questions. Table 4.2 shows the quality checklist formulated that consists of six quality questions.

Table 4.2: Quality checklist questions.

No	Questions	Answer QP
Q01	Is the aim is obviously stated?	Yes = 3, Partially= 1.5, No =0
Q02	Is the content of the study well expounded?	Yes = 3, Partially= 1.5, No =0
Q03	Does the study concentrate on the RP scalability issue?	Yes = 3, Partially= 1.5, No =0
Q04	Is the presented solution/technique clearly elaborated?	Yes = 3, Partially= 1.5, No =0
Q05	Is the assessment of the presented solution/technique performed on adequate project data sets or case studies?	Yes = 3, Partially= 1.5, No = 0
Q06	Is the result of the research evidently stated?	Yes = 3, Partially= 1.5, No = 0
Q07	Does the study elaborate on any factor that can cause the scalability issue?	Yes = 3, Partially= 1.5, No = 0

Each quality question has three possible answers associated with three quality points (QP) ('yes' answer with 3 QP, 'partially' answer with 1.5 QP, and

‘no’ answer with ‘0’QP). The procedure that we used in the application of the quality checklist is stipulated in a way that targets to assure the evaluation reliability and reduce the human biases selection as far as is possible. We aim to conduct the following procedure with the participation of both authors (student and supervisor) in evaluating the studies and selecting the primary studies:

- For each study of the chosen studies from inclusion and exclusion criteria, each author read the full text of the study.
- Each author evaluated the study on the basis of the formulated checklist by answering each quality question by assigning QP.
- The assigned QP for each study that had been read by the authors was collected.
- The discussion and comparisons among the authors were then taken place in meetings to handle the discrepancies with the aim of securing a consensus in assigning the final QP for each quality question and specifying the overall quality score of the study through summing all the QP of the listed quality questions.
- Finally, to ensure the reliability of the review’s studies that obtained a quality score less than 10.5 (which represents half of the full-quality score (21)) were only included.

After the execution of the quality assessment criteria, 22 studies have been finally selected as the primary studies. Table 4.3 depicts the selected studies with their associated reference numbers and total quality points.

Table 4.3: Quality Points of Selected Studies.

References	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Total quality points
[7]	1.5	3	3	3	0	0	1.5	12
[18]	3	1.5	1.5	1.5	1.5	1.5	1.5	12
[19]	3	1.5	1.5	3	3	1.5	3	16.5
[20]	1.5	1.5	3	3	1.5	1.5	1.5	13.5
[21]	3	3	1.5	1.5	1.5	1.5	1.5	13.5
[22]	3	3	1.5	1.5	1.5	1.5	1.5	13.5
[23]	3	3	1.5	1.5	1.5	1.5	3	15
[12]	3	3	3	1.5	3	3	3	19.5
[24]	3	3	1.5	1.5	1.5	1.5	1.5	13.5
[25]	3	3	3	3	3	1.5	3	19.5
[5]	3	3	1.5	1.5	1.5	1.5	1.5	13.5
[26]	1.5	1.5	1.5	1.5	1.5	1.5	1.5	10.5
[9]	3	3	1.5	3	3	3	3	19.5
[27]	3	3	1.5	1.5	1.5	1.5	1.5	13.5
[28]	3	3	1.5	3	3	3	1.5	18
[29]	3	1.5	3	3	3	1.5	1.5	16.5
[30]	3	3	3	3	3	3	1.5	19.5

Table 4.3: Quality Points of Selected Studies (continued).

References	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Total quality points
[31]	3	3	3	3	3	3	3	21
[32]	3	3	0	1.5	1.5	1.5	3	13.5
[33]	3	3	1.5	0	0	0	3	10.5
[34]	3	3	3	1.5	1.5	1.5	1.5	15
[6]	3	3	3	3	3	1.5	3	19.5

4.4 Data Extraction and Synthesis

In this SLR, the data extraction and synthesis were formed with the use of the google excel sheet and the Mendeley program. The data was extracted and synthesized based on the listed research questions. All the primary selected studies were critically analyzed to extract any relevant data for answering the stated research questions. To answer the RQ1, the data that is related to the causative factors of the scalability issue in RP and the meaning of each factor and impact degree were extracted and synthesized to reveal and analyze the factors that cause the scalability issue in the prioritization process. Concerning the RQ2, we extracted the data concerning the impact of each identified factor lead to the scalability issue during the prioritization process of a large number of requirements. To answer the RQ3, data related to the techniques proposed to address the scalability issue and their software development discipline applications were extracted. To elaborate on the RQ4, we extracted the features used by each of the identified techniques in solving the scalability issue and revealing their usage consequences and those are illustrated in tabular form. To answer RQ5, we extracted the limitations of the identified techniques and and provided future recommendations for further research to be conducted.

5

Results

This chapter presents the findings and discussion of this review study. It starts with providing an overview regarding the selected studies. Thereafter, a detailed discussion on the obtained answer of each research question is elaborated.

5.1 Overview of the selected studies

For this SLR, 22 papers were selected as the primary studies from different types of publication channels: 14 research papers were published in the Journal channels, five research papers presented in the conference proceedings, two research papers published in IEEE bulletins and one research paper retrieved from book chapter. Figure 5.1 depicts the percentage of the selected research papers shown: journal papers (64%), conference papers (23%), IEEE Bulletins (9%), and book chapters (4%).

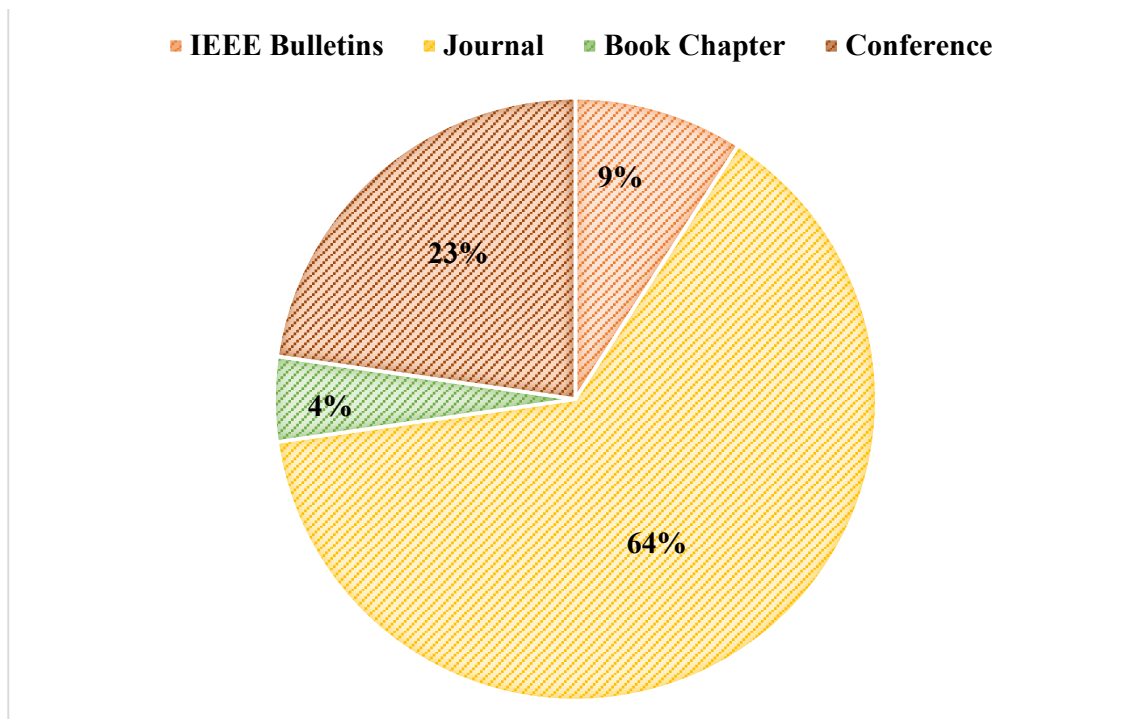


Figure 5.1: Publication Channels.

On the other side, Figure 5.2 depicts the publication years of the selected studies. As can be observed, the issue of the scalability was firstly considered in

1998 by Karlsson et al [12] in which the evaluation of the 5 existing techniques was conducted in terms of their ability in working with a large set of requirements. Starting from 2015 until 2020, there is a slight increase in the number of the papers that discussed the scalability issues, specifically in 2015, and 2019 and 2020, three to four studies were published. This can be related to the need for the fact that most of the existing software projects contain a large number of the requirements, which induce it to conduct more research in the RP field with concerns on the ability to prioritize a large set of requirements.

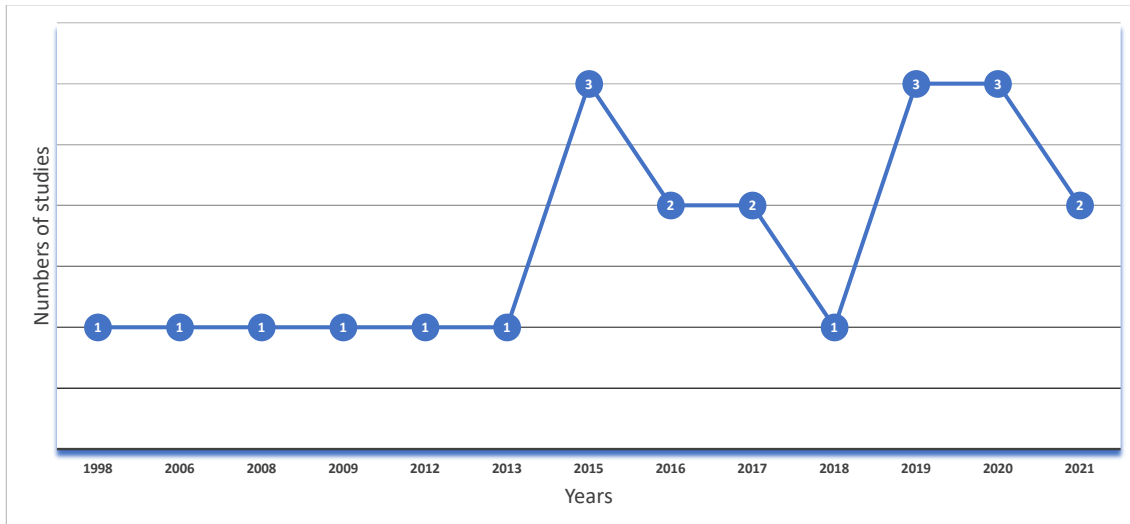


Figure 5.2: Publication years of selected primary studies.

5.2 RQ1. What are the causative factors of the scalability issue in RP and what is the meaning of each factor?

RQ1 is concerned with capturing the factors that induce to inability of the RP techniques in dealing with the large scale of requirements along with revealing the meaning of each factor. Table 5.1 presents the factors that raise the scalability issue in prioritization process along with their the meaning of each factor and citations for existing studies that reported each factor. Four factors are retrieved from the primary studies that reported at least one of them. These factors are number of comparisons, time, human efforts required and scarcity of the automation in conducting the prioritization process.

Table 5.1: Factors that considered for the scalability issues.

Factor Name	Citation	Description
Number of comparisons	[6], [9]–[12], [15], [19]–[22], [24], [26], [27], [29]–[31], [35]–[40]	Related to the total number of the pair wise comparisons that is required to produce the prioritized list of the requirements.
Time	[4]–[6], [9]–[12], [15], [19], [29], [32], [34], [36], [37], [39]–[41]	Has to do the time consumed in executing the prioritization process for large scale of requirements.
Scarcity of the automation	[6], [7], [10], [15], [22], [28], [36], [40]	Has to do with lack of providing automation process in prioritizing the requirements.
Human efforts required	[6], [11], [12], [21]–[23], [25], [30], [31], [33]–[37], [40], [41]	Related to the substantial efforts from the involved human (i.e stakeholders and experts) in prioritization process (i.e in specifying the priority value of each requirements , classifying the requirements, producing the ranked list of requirements).

5.3 RQ2. How do the identified factors lead to the scalability issue and which factors have the highest impact?

The first part of the RQ2 is aimed at revealing to what extent these identified factors induce to the scalability issue in prioritizing the large scale of requirements. The impacts of each identified factor towards the scalability issues can be elaborated and discussed as follows:

- **Number of Comparisons:** A comparisons of the relative priorities between pairs of requirements is one of the common ways for conducting RP process in most of the existing techniques such as bubble sort [14], AHP [12], [14], and pairwise comparison technique [10], [12], [42]. The number of comparisons increases dramatically as the number of requirements grows, which makes the prioritization process to be highly tiring and leads to introduce human mistakes with respect to the performing high number of comparisons. This will undoubtedly lead to the scalability issue of the prioritization process [10], [11], [35], [40], [43]. For instance, AHP and pairwise comparison technique have

been proven to have scalability as main issue in their prioritization processes due to the high growth in pair-wise comparisons that exist in handling the large set of requirements. With 121 requirements, the number of comparisons for AHP need is 7260 based on the define formula $n*(n - 1)/2$, which make the prioritization process to be tiring tedious.

- **Time:** Time is ordinarily critical in the industrial side, which makes it a crucial variable for the evaluation and implication of requirement prioritization techniques [11], [44]. Time is related to the time consumed in executing the prioritization process for large scale of requirements, starting from the time for initiating the prioritization process to producing the ranked list of requirements. The scalable RP technique should work with large set of requirements without being time consuming in conducting RP process [41]. Hence, time is considered as a factor that can induce the scalability issue, since the scalability of the technique is impacted by the time consumption for producing the prioritized list of requirements. Thus, as the time consumption grows when of the number of requirements increases due to the increased complexity in producing an the prioritized list, more work will be required [12]. For instance, when the number of requirements is doubled in a list (large set of requirements), RP techniques will require more effort in terms of time for prioritizing the requirements such as AHP and bubble sort techniques, which will consume much time to perform the pairwise comparisons for each requirement. This makes the technique to be impracticable in the industry side in terms of using it in prioritizing large number of requirements [12], [39], [41].
- **Scarcity of the automation:** Most of the existing RP techniques lack automated system [10]. Prioritizing large number of requirements without introducing automation and intelligence to the prioritization process can impact the efficiency of the technique [10], [40]. The prioritization process can be more complex to be conducted due to manual process that will be needed to perform the computational calculation for identifying the relative priority value of each requirement. This manual computational complexity may result in error in generating the final approximation ranking of the prioritized [10], [40]. Hence, this makes RP technique unable to deal with hundreds or thousands of requirements [10], [40]. However, the fact of introducing the auditioned the predominant notion is that automation and intelligent process support of prioritization will enhance the scalability because some of the effort can be turned to the automation process [40].
- **Human efforts required:** Most of the existing RP techniques are heavily relied on the human involvements for conducting the prioritization process [31]. Human efforts required refers to heavy reliance on human expertise participation to conduct and the RP process in terms of inputting the preference information on the requirements to be prioritized, specifying the priority value of each requirement or classifying the requirements [6], [9], [10], [15], [40]. For instance, when we dealing with large set of requirements, AHP technique will

require each involved stakeholder to define priority value for each requirements and perform the pairs pairwise comparisons process. This will consume much time in obtaining the inputs and make activity to be counterproductive and difficult to manage if the human expertise shortage occur. In addition, the reliability of the exiting techniques could be challenged if there is an over reliance on human expertise participation in the RP process which is not encouraged [6], [9], [15], [16]. Thus participation of human expertise can be challenging while utilizing the techniques and may lead to the possibility to make the faulty results while implementing the exiting techniques in a wrong way [6], [9], [15], [16]. Furthermore, biases introduced by the participation of human expertise during the prioritization process to prioritize the requirements can affect the quality of the results in establishing a quality software system.

Concerning the second part of RQ2 that aims to reveal the significance degree of the listed factors in raising the scalability issue in RP process, Figure 5.3 depicts the significance degree of each identified causative factor. The significance degree of each reported factor is utilized to reveal the factors which are considered as the most important causative factor for inducing the scalability issue in the prioritization process of the requirements. The significance is measured by counting the usage frequency for each listed factor from the selected primary studies. The usage frequency is related to the number of times each factor is reported or mentioned by selected studies as causative factor induce to inability of dealing with large scale-requirement in prioritisation process: number of comparison (48%), time (28%), human efforts required (involvement) (17%), and Scarcity of the automation (7%).

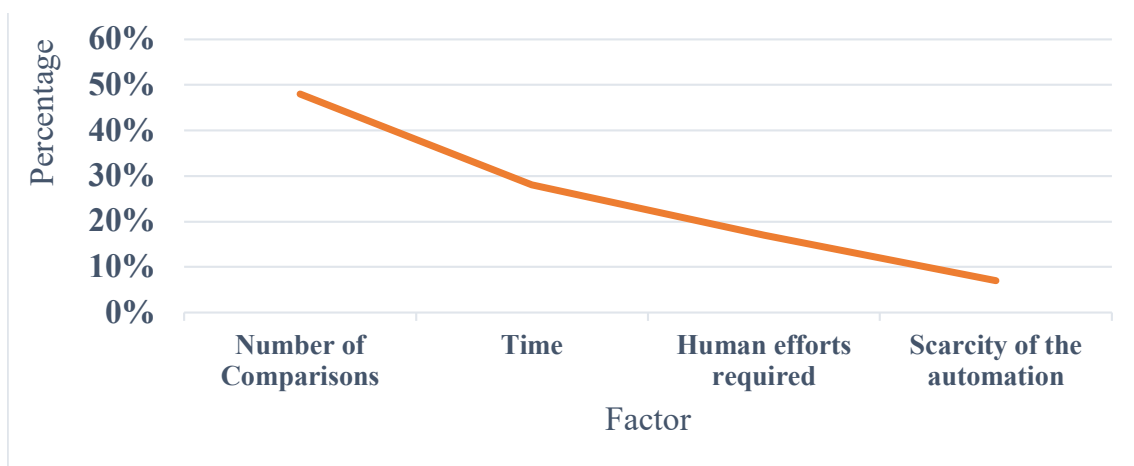


Figure 5.3: Importance degree of Scalability Causative Factors.

On the basis of the revealed importance degree, the number of comparisons factor is reported to be more prime causative factor as compared to other causative factors. This finding can be related to the concern of the majority of the related studies on catering the scalability issue by reducing the number of comparisons. With minimising the number of comparisons, the execution of the RP process can be implemented on large number of requirement without requiring high human efforts required and consuming large amount of time , which resulting to be able to deal

with large scale of requirements as postulated by most of the studies.

5.4 RQ3: What are the techniques that have been proposed to address the scalability issue in the RP process and what are their software development usage contexts?

The aim of this question is to identify and describe the existing techniques that have been proposed to handle the scalability issue, along with analyzing these identified techniques in terms of their software development usage contexts. From the selected studies, 23 techniques have been identified that have been proposed with aim of addressing the scalability issue.

Lunarejo et al [7] presented a semi-automatic technique that deals with the scalability issue by implementing a deep learning algorithm. Abdelazim et al [18] presents a generic framework to be utilized in addressing the scalability with considering dependencies among the requirements. This framework suggested to integrate the existing RP techniques into a comprehensive model of prioritization to address the issue of scalability in the agile development process. In addition, Achimugu & Selamat et al [19] proposed a hybridized technique for prioritizing large set of requirements based on K-Means and Evolutionary Algorithms, which comprises requirements with relative weights of stakeholders.

Meanwhile, Beg et al [20] mentioned an technique using B-Tree to prioritize large set of requirements. While, Perini et al [21], proposed a technique called Case-Based Ranking (CBRank). This technique exploits machine learning techniques to overcome the scalability problem by reducing the human efforts required in the prioritization process. The CBRank method succeed to handle the scalability issue and reducing the acquisition effort by combining human preference elicitation and automatic preference approximation. Shao et al [22] introduced a technique called DRank based on a machine learning algorithm named RankBoost to reduce human efforts, which assists to address scalability. Chandra et al [23] proposed a technique called Goasrep. Goasrep is used to prioritize the large scale of requirements using a goal-oriented approach and then AHP used to. The function point analysis approach (FPA) and the COCOMO model were adapted to estimate the cost and efforts. Testing with large set of requirements is needed to generalize the performance of this technique in addressing the scalability.

J. Karlsson et al [12] have conducted an experiments on the performance of the five techniques. The results demonstrated that the Hierarchy AHP and Minimal Spanning Tree can deal with large set of requirements. Hierarchy AHP reduces the required number of pairwise comparisons and redundancy, and the minimal spanning tree approach was reported to be fast in prioritizing the large set of requirement

comparing to the other techniques due to its ability minimizing the of pairwise comparisons.

Meanwhile, Berander et al [24] conducted an experiment and evaluated this experiment in two different ways of priorities using hierarchical prioritization method. Priorities are either calculated with or without compensation for the size of each block of requirements and the results demonstrate that hierarchical cumulative voting (HCV) is scalable and can deal with large number of requirements.

Ibriwesh et al [30] proposed a technique namely ReDCCahp to overcome the scalability issue by reducing required number of pairwise comparisons. The performance of the ReDCCahp technique was evaluated in terms of addressing scalability comparing to the AHP. The results demonstrated that ReDCCahp technique is more scalable than AHP in prioritizing large number of requirements with lower effort. On the another research Ibriwesh et al [25] proposed a technique based on multiple perspective prioritization algorithm (MPPT). A controll experiment was conducted to compare the MPPT technique with two existing techniques: AHP and Wieggers. The results demonstrated that MPPT technique is more scalable than AHP and Wieggers by reducing the number of pairwise comparisons.

Nidhra et al [5] proposed a technique named NAcAHP based on the combination of two existing techniques: the numerical assignment technique and the AHP technique. In this techniques, the numerical assignment utilized to classify each requirement into groups. Later, the AHP technique is applied to prioritize those requirements selected in each group. The performance of ‘NAcAHP’ was compared with AHP in terms of time and scalability. The results showed that proposed technique minimize the time and complexity of pair wise comparisons which lead to handle the scalability.

In addition, Ayub et al [26] proposed a method called PGAHP, which combines planning games, AHP technique. The author applied AHP to planning games and then resolved the problem of two or more requirements having the same priority. The evaluation results demonstrated that while dealing with large number of requirements, PGAHP reduce the complexity of prioritizing the requirements by minimising the pairwise comparisons as compare to AHP.

Babar et al [9] mentioned a technique named Priority Handler (PHandler). To handle the scalability issue, the back-propagation neural network is used to predict the value of a requirement to reduce the extent of expert biases and make the PHandler efficient in addressing the scalability issue. The evaluation results demonstrated the ability of technique to work with large scale of requirements. However, this techniques is highly relied on the involvement of the expertise to conduct the process of the prioritization and lack of addressing the dependencies among requirements. Tufail et al [27] conducted comparative analysis on ten existing techniques in terms of scalability, complexity, accuracy, suitable size of the requirements sets and handling risk factor. Moreover, strengths and weaknesses of those ten exiting

techniques were analyzed. The analysis demonstrated that planning game (PG) technique has the ability to dealing with large number of requirements.

Duan et al [28] proposed technique called Pirogov. This technique uses data mining and machine learning techniques to prioritize large set of requirements. The prioritization has been conducted according to stakeholder interests, business goals, and concerns such as safety or performance. The requirements are automatically clustered according to different goals, such as feature sets, business goals, or high-level use cases. Stakeholders then prioritize the clusters manually using one of the traditional prioritization techniques. The system then automatically generates a list of ordered requirements. The evaluation results shows the ability of the technique in addressing the scalability by adopting the data mining method. However, this technique still faces the issues of the incapability to address requirement interdependencies, over reliance on the involvement of human expert.

ReproTizer is a technique proposed by Achimugu et al [29] to prioritize large set of requirements. This technique used the weight decision matrices to determine the weight vectors of requirements with an aggregation operator (AO), which computes the global weights of requirements. The evaluation results present the ability of techniques to deal with large set of requirements. However, this technique is not able to handle the requirements interdependencies in the prioritization process. In addition, Lim & Finkelstein et al [31] proposed a technique called Stakerare for prioritizing large set of requirements. This techniques used a social networks and collaborative filtering to identify and prioritize requirements in large software projects. The evaluation of this technique was conducted with industrial project called RALIC that includes large set of requirements. The findings presented ability of the technique in working with large set of requirements. Heavily reliance on the experts participation along with not considering dependencies among the requirements in the prioritization process are revealed as main limitation of this technique.

Another technique introduced by McZara et al [32] is called SNIPR which assist in requirements prioritization and selection based on natural language processing and satisfiability modulo theories (SMT) solvers. The results demonstrated that SNIPR consumes less time in prioritizing large set of requirement by reducing the pairwise comparisons due to the usage of the SMT.

Asif et al [33] proposed a SAFFRON technique that can solve the rank reversal problem during the prioritization of large set of requirements. This technique uses collaborative filtering techniques to resolve the rank reversal issues and decrease the number of interactions with stakeholders. The evaluation of this technique is implemented and tested with the RALIC data set and the result shows that proposed technique reduces human interaction while updating prioritized requirements list.

Additionally, Yaseen et al.[34] introduced a technique called SAHP, which uses AHP and spanning trees. AHP is used to handle the dependencies among the requirements and requirements are statistically comprise as pairwise to reduce the

dependency of requirements. In addition, spanning tree used to reduce the waiting time of requirements by equal and efficient distribution of requirements. The result demonstrated that proposed technique is able to reduce the pairwise comparisons by minimizing the dependency between the requirements.

Furthermore, Hujainah et al [6] proposed the technique named SRPTackle. This technique introduced a new semi-automated process for handling the scalability issue based on a combination of a constructed requirement priority value formulation function. Using a multi-criteria decision-making method (i.e., weighted sum model), clustering algorithms (K-means and K-means++), and a binary search tree minimize the need for expert involvement and increase efficiency.

Furthermore, the second of this question aims to identify the usage context of the identified techniques. Usage context is refer to the technique has been proposed for or the context with the techniques has been implemented in. In RP there are certain number of usage contexts that have been revealed by Hujainah et al [15].

Based on RP usage contexts reported in Hujainah, we analyzed each technique of 23 identified techniques in terms of the usage contexts this SLR. We found that each technique has only one usage context, and several technique belongs to one usage context. We found 12 usage context of the identified techniques: Software Product-Lines paradigm (SPL), Agile Software Development (ASD), Value-Based Software Development (VBSD), Software Release Planning (SRP), Exploit Domain knowledge (EDK), social network system development (SNSD), Solution-Oriented Software Requirements (SOSR), Real-Client Custom Development projects (RCCD), Commercial Software Systems Development (CSSD), Market-Driven Software Development (MDS), Goal-Oriented Requirement Engineering (GRQE) and Not Specified (NS) context (e.g., where RP techniques do not mention any context in the paper.)

Figure 5.4 visualize the percentage of usage context mentioned. The percentage were measured by how many techniques were used or proposed in/for the identified context . As can be observed from Figure 5.4, most of the RP techniques that addressed the scalability issues are focused on the SRP context with (33% percentage). MDS, NS, and RCCDP contexts have a percentage of 25%, followed by SNSD, VBSD, GRQE, and ASD contexts with 17%, and finally SOSR, CSSD, EDK, and SPL with of 8%. SRP context is the highest usage context in scalability issues. This can be related to the reason that most of the software development organizations used the prioritization process in agile environment for handling the large scale of requirements on the basis of release planning.

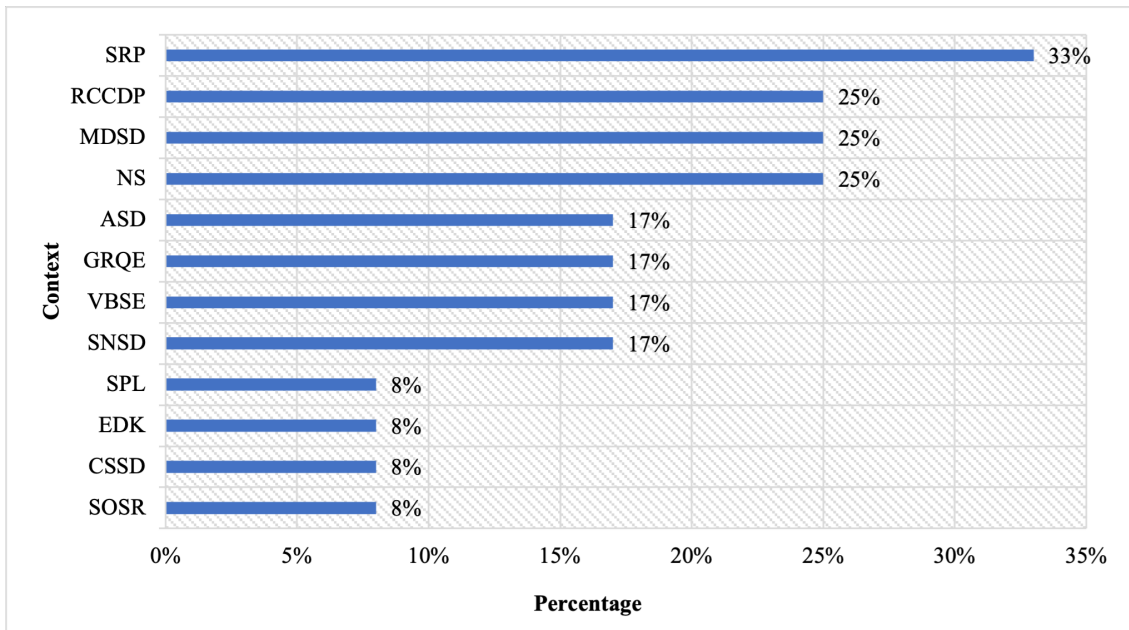


Figure 5.4: Percentage of Used Contexts in the Selected Studies.

5.5 RQ4: What are the features (e.g., type of the adapted method, algorithm) used by the identified RP techniques in RQ3 to handle the scalability issue and what are their usage consequences in solving the scalability issue?

The aim of this question is to reveal features (e.g., type of the adapted method, algorithm) used by the identified techniques in solving the scalability issue and revealing their usage consequences in addressing the scalability. Usage consequence here refers to the impact of using the feature in making the technique able to handle the scalability. Table 5.2 presents the outcome of analyzing each identified technique in terms of the used feature and their usage consequences based on the conducted analysis, we categorized the identified techniques based on their feature used in handling the scalability issues as depicted in Figure 5.5: Machine learning based techniques (techniques that adapt any machine learning and AI algorithms), and combination of the existing based techniques (techniques that adapt combination of the existing methods for prioritizing large set of requirements).

As observed from Figure 5.5, most of the existing techniques adapted the Machine learning algorithms in addressing the scalability issue. Back propagation network was adapted in the Phandler technique [9]. The back-propagation neural network was used to predict the value of a requirement in order to reduce the extent of expert biases and make the Phandler efficient, which lead the technique to be able to work with large set of requirements.

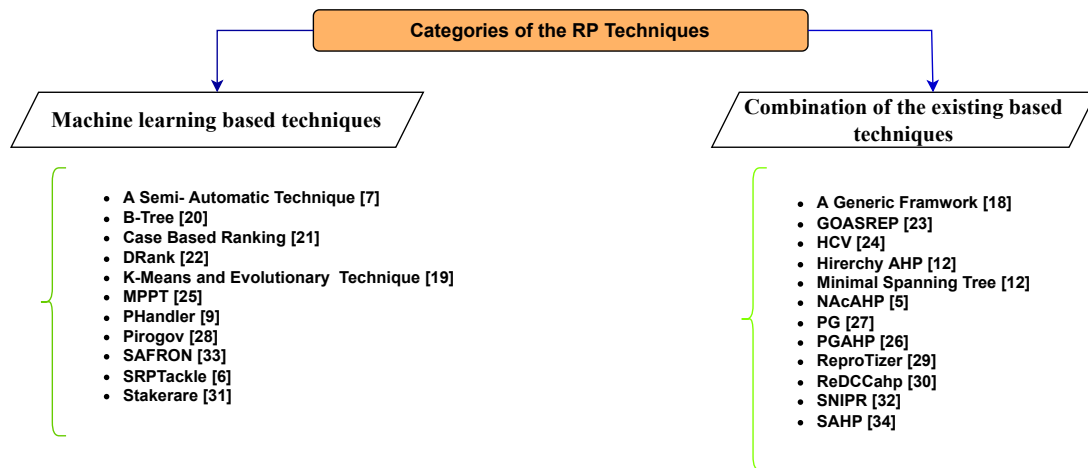


Figure 5.5: Categories of the RP techniques based on the features used in addressing the Scalability issue

DRank technique adapted RankBoost algorithm in prioritizing large set of requirements with considering the dependencies among requirements [22]. The RankBoost was applied to specify and generate the prioritization according to stakeholder preferences [22]. The implementation of the RankBoost lead to reduce the difficulty of evaluating the prioritization, and human efforts needed in prioritizing the large set of requirement as compared to AHP technique that requires more efforts in conducting the pairwise comparisons for conducting the prioritization process.

A semi-automated technique adopted machine-Learned ranking algorithm in prioritizing the large set of requirements. [7]. Machine-learned ranking was applied to improve the requirement’s prioritization in the software product line(SPL) projects [7]. In this technique, the machine-learned ranking was implemented to reduce the effort required from stakeholders participation in prioritizing the large set of requirements, which induce to make technique able to work in large set of requirements with less efforts and time.

In addition, in the CBRank technique [21], the Case Based Ranking algorithm has been adapted, in which CBRank method reduces the efforts of the involved human participation through the approximate combination of human choice expression and automated choice, which led to make the technique able to work with large set of requirements.

Additionally, the combination of K-Means and Evolutionary Algorithm were used as features by the Achimugu & Selamat et al [19] in addressing the scalability issue. The K-Means and evolutionary algorithms was used in defining the relative weights of participants and minimisation of cluster performance indexes, square error and error criteria. This lead to define the priority value of each requirements without performing less number of pairwise comparisons, which induces to make this technique able to work in large set of requirements.

Furthermore, the Pirogov adopted Spherical K-means technique as a feature to prioritize requirements according to stakeholder's interests, business goals, and cross-cutting concerns [28]. Spherical K-means was applied to prioritize the requirements according to the stakeholder preference [28]. Implementation of Spherical K-means technique assists to reduce the number of pairwise comparisons which lead to the scalability of the proposed technique.

While in the Stakerare technique, social networks and collaborative filtering approach were adapted to reduce the efforts need in prioritizing the large scale of requirements [31]. However this technique was unable to address the rank update. Thus, SAFFRON was proposed by Asif et al [33] to solve the rank reversal issues and decrease the number of interactions with stakeholders using the uses collaborative filtering technique.

Another technique named multiple perspective prioritization (MPPT), adapted MPPT algorithm as a feature to address the scalability issue [25]. The MPPT succeed to handle the scalability by consuming less time, reducing the need for human participation efforts of, comprehensive logical answers, and giving more control to complete the task. Additionally, SRPTackle [6] succeed to handle the scalability issue by using the combination of a new constructed requirement priority value formulation function, and clustering algorithms (K-means and K-means++), along with a binary search tree. The evaluation results demonstrated ability of the technique in addressing the scalability by reducing the time consumption and number of comparisons along with minimizing the need for expert involvement and increasing efficiency. Furthermore, Beg et al [20] used B-Tree Algorithm. The usage of the B-Tress lead to reduce the number of comparisons needed as it is considered branching factors in prioritization and when the branching factor is increased, the number of comparisons required is drastically reduced.

Meanwhile, other existing techniques proposed some features in addressing the scalability based on a combination of existing RP techniques. The Chandra et al [23] presented GOASREP technique based on the goal-oriented approach and COCOMO model. Those two features used to estimate the cost and effort by using function point analysis approach. In addition, the NAcAHP technique adapted two techniques: numerical assignment and AHP [5]. The numerical assignment technique first categorizes each requirement into groups, and later AHP technique is applied to prioritize those requirements selected in each of the groups.

Similarly, Ayub et al [26] proposed a technique namely, PGAHP which used two techniques called Planning game (PG) and Analytical hierarchy process (AHP). The PGAHP reduces the complexity and memory of pairwise comparison by applying AHP and PG. Planning games divide the requirements into three categories which reduces the complexity. SNIPR [32] uses natural language processing (NLP) and satisfiability modulo theories solvers (SMT). The use of NLP allows for identification of some of the requirements dependencies. Variations in the priority

of the requirements output by the SMT solver are resolved by iterative pairwise comparisons and then used as updated input to the SMT solver. The AHP is used to re-rank a subset of the requirements for increased accuracy. Whereas, the SAHP used AHP and spanning trees as a combination to overcome the scalability issues in which it is compare the time complexity of the SAHP with the existing AHP [45].

In addition, the a generic framework integrated RP techniques (INVEST SMART, Work breakdown structure, Story point, MoSCoW, Kano) into a comprehensive model of prioritization to address the issue of scalability using four generic steps [18]. Where first step is identification: identify stakeholder, feature, and user story. The second step is Verification: verify the user story. The third step is Estimation: estimate size, effort and value of the user story. The fourth step is Prioritization: prioritize user stories. This framework can handle the changes required at any stage of the development cycle in the context of the agile development process.

Furthermore, In Hierarchy AHP technique, the hierarchical structure and AHP methods were used to handle the scalability [12]. Using a hierarchical structure lead to reduce the required number of decisions, and the amount of redundancy. Thus it is more sensitive to judgmental errors than AHP technique. On the other hand the minimal spanning tree approach is adapted by Karlsson et al [12] due to its ability in reducing the number of pairwise comparisons dramatically, which made it to be fast in doing the prioritization for large set of requirements as compared to AHP technique.

Meanwhile, In Hierarchical Cumulative Voting (HCV), the AHP and commutative voting methods were adapted to address the scalability issue [24]. HCV solve the complex multi criteria decision problems by arranging the problem into the hierarchies. The HCV method solving the scalability problem by introducing compensation for block size and it's also reduces the number of comparisons by using the hierarchical features of HCV respectively. As well as Tufail et al [27] uses PG as a method in which requirements are prioritized separately and are not compared to other requirements, it scales well with a large number of requirements. Additionally, In the ReDCCahp [30], dynamic consistency checking algorithm were adopted to address the scalability. The usage of consistency checking algorithm assists to reduce the number of pairwise comparisons by removing the number of redundant comparisons. In the ReproTizer technique [29], weight decision matrices was used to determine the weight vectors of requirements, along with with an aggregation operator (AO), which used to compute the global weights of requirements. The evaluation results present the ability of this technique in dealing with large set of requirements.

Table 5.2: Analysis of Identified techniques in terms of the features used and their usage consequences.

Name of the Techniques	Features Used	Usage Consequence
A semi-automatic technique [7]	machine-learning ranking and a deep learning algorithm	For reducing the manual effort of the stockholder's author used a machine learning algorithm.
A generic framework [18]	a generic steps : Identification: identify stakeholder, feature, and user story. Verification: verify user story. Estimation: estimate size, effort and value of the user story. Prioritization: prioritize user stories.	This generic framework used some generic steps by which it's easier to handle the changes required at any stage of the development cycle in the context of the agile development process.
B-Tree [20]	B-Tree Algorithm	The B-Tree algorithm uses a branching factor for prioritization and increasing the branching factor significantly reduces the number of comparisons required.
Case-Based Ranking [21]	Case-based ranking strategy	The CBRank method reduces user effort by an approximate combination of human selection and automatic selection.
DRank [22]	Machine learning method called RankBoost	DRank uses a machine learning method called RankBoost helps to combine the prioritization of different dimensions which reduces human effort.
GOASREP [23]	goal-oriented approach and COCOMO model	The Function Point Analysis (FPA) approach estimates the cost of each requirement and uses the COCOMO model to estimate the effort of each requirement.
Hierarchy AHP [12]	Hierarchy AHP	Using a hierarchical structure reduces the number of decisions required and the amount of redundancy.

Table 5.2: Analysis of Identified techniques in terms of the features used and their usage consequences (continued).

Name of the Techniques	Features Used	Usage Consequence
Hierarchical cumulative voting (HCV) [24]	Block size	The HCV method solves the scalability problem by introducing compensation for block size and it also reduces the number of comparisons using HCV's hierarchical features.
K-Means and Evolutionary Technique [19]	K-Means and Evolutionary Algorithms	K-Means and evolutionary algorithms that include requirements with relative weighting of participants and minimization of the cluster performance index.
Multiple perspective prioritization [25]	MPPT algorithm	MPPT consumes less time and lead to reduce the licitation effort.
Minimal spanning tree [12]	Minimal Spanning Tree	the minimum spanning tree method is very fast to reduce in pair-based comparisons.
NAcAHP [5]	Combine the numerical assignment technique with the AHP	NAcAHP has reduced the time and complexity by performing pair-based comparisons. The numerical assignment strategy first classifies each requirement into groups, and then the AHP strategy is applied to prioritize those requirements selected in each group.
PGAHP [26]	Combination of Planning game(PG) and Analytical hierarchy process (AHP)	PGAHP method helps to reducing the complexity by PG through the divination of requirements into three sections and reduce the pairwise comparison by applying AHP.
PHandler [9]	(back-propagation) network and analytical hierarchical process	Back-propagation neural network is used to reduce the amount of bias and to estimate the value needed to make PHandler efficient.
PG [27]	Planning Game method	PG helps to reducing the complexity by divination of requirements into three sections. Since each requirement is prioritized separately and is not comparable to other requirements, it measures well with a large number of requirements.

Table 5.2: Analysis of Identified techniques in terms of the features used and their usage consequences (continued).

Name of the Techniques	Features Used	Usage Consequence
Pirogov ([28])	Spherical K-means (SPK)	Data mining and machine learning techniques helps to prioritize needs according to stakeholder interests, business goals and cross-cutting concerns such as safety or performance requirements. The number of comparisons with the number of requirements increases rapidly.
ReproTizer [29]	a WS (Weight Scale) aggregation operator (AO)	As the weight scale project gains the ability to perceive the impact of stakeholders, there may be different weight requirements on the final outcome. Weight scale with aggregation operator (AO) that calculates the global weight of requirements. ReproTier creates a better response time with less complexity.
ReDCCahp [30]	dynamic consistency checking algorithm	Dynamic consistency checking algorithm reduce the number of pairwise comparison when the number of unnecessary comparisons is removed, the number of comparisons will also decrease.
Stakerare [31]	social networks and collaborative filtering and recommendation.	social networks and collaborative filtering reduce the efforts need in prioritizing the requirements.
SNIPR [32]	natural language processing and satisfiability modulo theories solvers.	NLP can be used to identify some dependencies of a requirement. The deviation in the priority of the SMT solver's output of the request is resolved by an iterative pairwise comparison and used as an updated input to the SMT solver. AHP is used to reclassify a subset of requirements for greater accuracy.
SAFFRON [33]	collaborative filtering technique	To resolve the rank reversal issues and decrease the number of interactions with stakeholders.

Table 5.2: Analysis of Identified techniques in terms of the features used and their usage consequences (continued).

Name of the Techniques	Features Used	Usage Consequence
SAHP [34]	SAHP, which uses AHP and spanning trees in combination.	comparing the time complexity of the SAHP with the existing AHP by which, the author tries to overcome the scalability issues.
SRPTackle [6]	Combination of a constructed requirement priority value formulation function. Using a multi-criteria decision-making method (i.e., weighted sum model), clustering algorithms (K-means and K-means++), and a binary search tree.	reduce the time consumption and number of comparisons along with minimize the need for expert involvement and increase efficiency.

5.6 RQ5. Which of the RP technique from RQ3 have been evaluated in terms of the scalability issue, and to what extent these techniques have covered the identified causative factors of scalability issue in their evaluation processes?

The aim of this research question is to assess to what extent the identified RP techniques from the RQ3 (that focused on handling the scalability issue) have been validated and addressed the identified causative factors of the scalability issue in their evaluation. To measure the evaluation extent, we have analysed each identified technique with respect to being evaluated or not, the evaluation environment settings in terms of revealing the name of the used projects/data sets in the evaluation, and types of the evaluation environment settings with respect to academic settings (the source of the used projects /data sets are from the academic institution or contexts, and industrial settings (source of the used projects/data sets in evaluation from the industry organizations or contexts), and revealing the causative factors being considered in addressing the scalability issue. The outcome of this analysis is presented in Table 5.3.

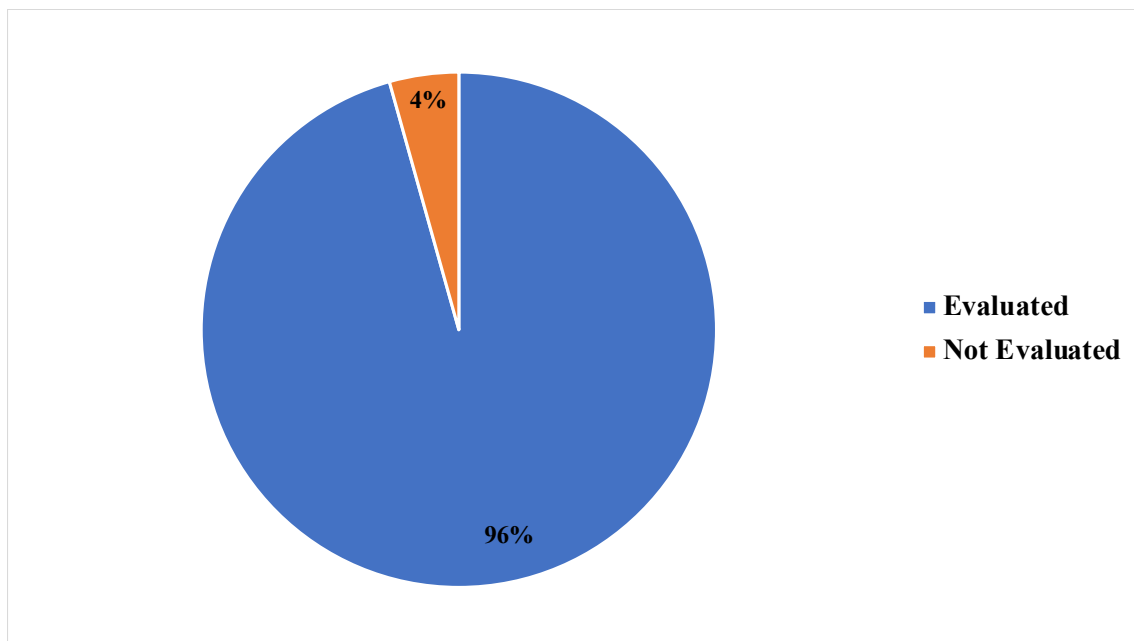


Figure 5.6: Percentage of the evaluation methods in evaluating the existing technique.

Figure 5.6 presents the percentage of the evaluation extent of the identified techniques with respect to being evaluated or not. The percentage is calculated based on the number of techniques being evaluated or not. It can be observed that, out of 23 identified techniques, 22 are evaluated which refers to 96% percent, and

only 1 technique (which refers to 4% percent) that presented by AbdElazim et al [18] was not evaluated due to this technique was introduced in the form of a generic framework that can be implemented in addressing the scalability issue. These results indicate that the strong interest of researchers in not only proposing a technique but also in evaluating the performance of these techniques in terms of addressing the scalability issue.

Figure 5.7 presents the evaluation methods used in assessing the performance of the identified RP techniques. Experiment and case study are found to be the evaluation methods used in evaluating the identified RP techniques. In this research, we adopted The definitions of experiment and case study from [46] in to differentiate between the meaning of these terms. The term of the case study indicates to evaluation manner that is conducted based on monitoring real-time projects, assignments, and feedback from industry/companies, whereas experiment refers to evaluation method that takes place in a controlled environments.

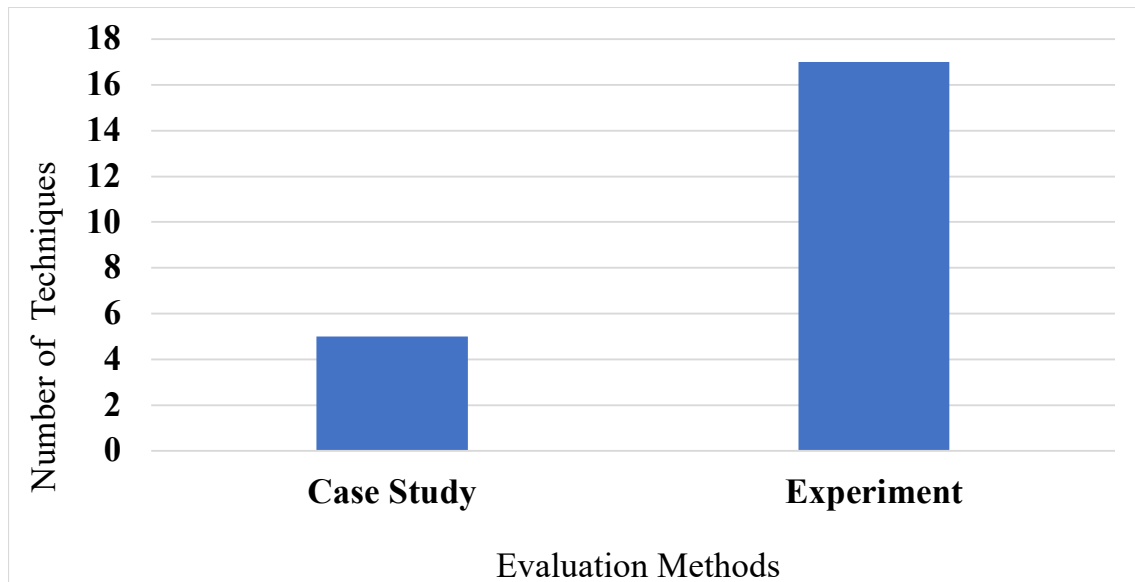


Figure 5.7: Evaluation methods in evaluating the existing technique.

As shown in Figure 5.7, most of the empirical studies conducted experiments in evaluating the performance of the identified RP technique. Out of 22 evaluated techniques, 17 techniques have been evaluated using experiments: A semi-automatic technique [7], B-Tree[20], K-Means and Evolutionary Technique [19], Drank [22], Hierarchy AHP [12], hierarchical cumulative voting [24], Minimal spanning tree [12], multiple perspective prioritization algorithm [25], NAcAHP [5], PGAHP [26], Phandler [9], Pirogov [28], ReproTizer [29], ReDCCahp [30], Stakerare [31], SNIPR [32], SRPTackle [6]. Among these seventeen experiment studies, four (4) studies were conducted in academic settings, and the rest 13 studies were conducted in industrial settings. This can related to the availability of benchmark datasets of the real industrial projects that used in assessing the scalability performance of the RP techniques in terms of the scalability.

Additionally, Figure 5.8 depicts the evaluation environment type (academic or industrial setting) in terms of the used evaluation environment context (e.g., source of the used projects or benchmark dataset), involved subject type (if any). As can be noticed from Figure 5.8, 18 identified techniques were conducted in the industrial settings with a percentage of 78%, and 18% (4 techniques) were evaluated in academic settings. only 1 study (4%) ReproTizer [29] conducted in both academic and industrial settings, which has been evaluated using the industrial datasets and academic projects. We have observed that the evaluation of all five techniques that were evaluated using case studies were conducted in industrial settings. While, among 18 techniques evaluated by experiments, only 4 techniques were conducted in academic contexts and the rest were in industrial contexts.

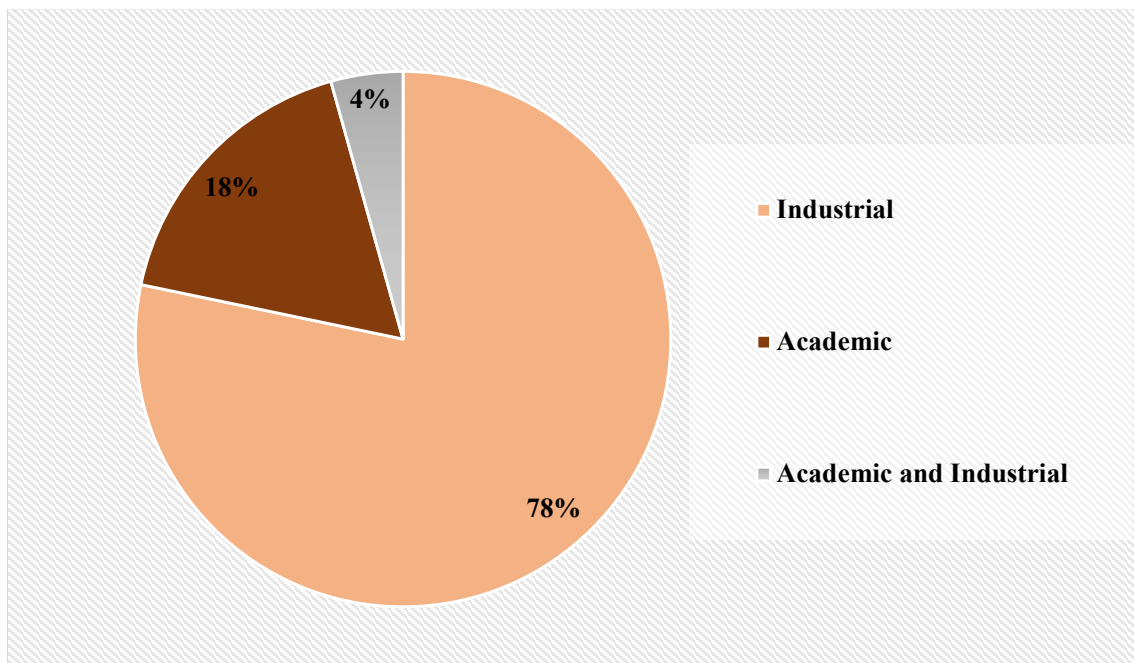


Figure 5.8: The percentage of the evaluation environment settings used in assessing the performance of the existing techniques.

The evaluation industrial settings yielded the highest percentage as the most of the techniques was evaluated using industrial projects or data sets. This results are in line with the need to have a large number of requirements that can mostly be found in industrial contexts as most of the current system projects include a large number of requirements which are necessary to evaluate the capability of technique in addressing the scalability issue. Also, the existence of benchmark data sets from actual industrial projects assists most of the existing techniques to be used in evaluating how well approaches scale. Additionally, the RALIC (stands for real Replacement Access, Library and ID Card software) benchmark data set is observed to be among the pioneer data sets that have been used in the evaluation of identified RP techniques with respect to the scalability issue. This can be related to the reason that the RALIC benchmark data set project is considered to be one

of the first available and most complete data sets that include a large number of requirements for the RALIC industrial software project [6].

Based on analysis conducted in Table 5.3, Figure 5.9 depicts the ratio of considering each identified causative factor in the evaluation of the identified techniques. Out of the 22 evaluated techniques, the number of comparisons factor has been considered in the evaluation of 13, the time and human efforts required factors considered in the evaluation of 9 and 8 techniques, respectively, and finally the scarcity of the automation factor was considered in evaluation of 4 techniques. Here, the number of comparisons factor were associated with the highest rank of considered factor in the evaluation of the identified techniques with respect to addressing the scalability issue. This can be related to the reason that most of the identified RP techniques were proposed to address the scalability issue by minimizing the number of pairwise comparisons that dramatically increased in prioritizing the large set of requirements. However, there is still lacks in providing empirical evidence of the ability in addressing the scalability issue by only reducing the number of the comparison without the need of considering other causative factors such as human efforts required and scarcity of automation.

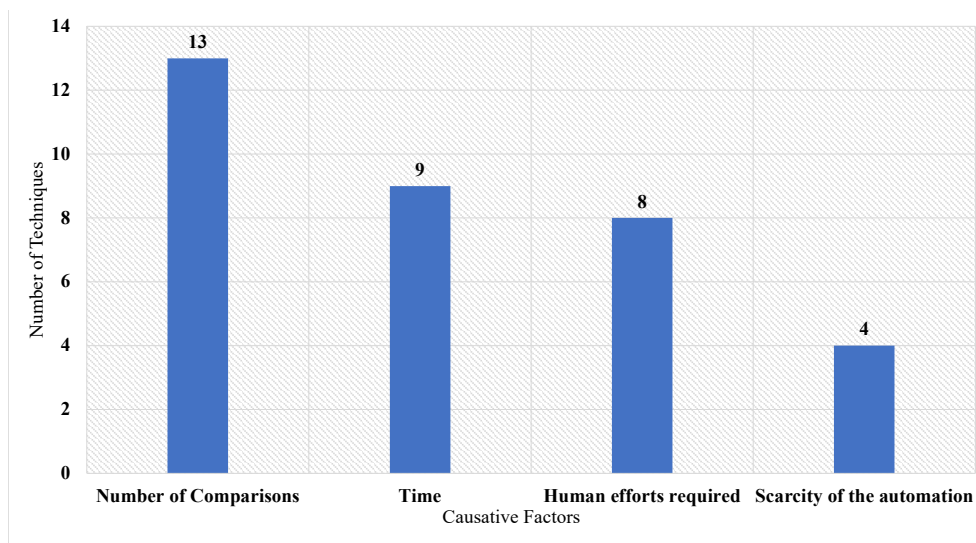


Figure 5.9: Percentage of the causative factors being catered in addressing the scalability issue in all identified techniques.

In addition, Figure 5.10 presents the number of the causative factors being considered by each identified technique. Out of 22 evaluated techniques, only one technique (SRPTackle) [6] considered all four factors (human efforts required, time, number of comparisons, scarcity of the automation) during its performance evaluation. Although, it is applicable to consider all the four factors in the evaluation as presented in the SRPTackle technique, we noticed that 10 techniques considered 2 factors in their evaluation: Semi-Automatic Method [7], Case-Based Ranking (CBRank) [21], Drank [22], Hierarchy AHP [12], K-Means and Evolutionary Technique [19], Minimal Spanning Tree [12], Multiple Perspective Prioritization

Technique [25], ReproTizer [29], SAHP[34]. While the rest of the 11 techniques considered only 1 factor in their evaluation: B-Tree [20], GOASREP [23], HCV [24], NAcAHP [5], PGAHP [26], Phandler [9], PG [27], Pirogov [28], ReDCCahp [30], Stakerare [31], SNIPR [32], SAFFRON [33]. This finding indicated that most of the identified existing techniques have not sufficiently covered the causative factors in their evaluations as at most up to 2 factors have been considered (mostly attributed to the fact that they are primarily considered as proof-of-concepts) except the evaluation of SRPTackle technique that consider the four factors. Thus, a comprehensive evaluation of the performance of these identified techniques is required to ensure their ability to adequately address the scalability issue by considering the four identified causative factors.

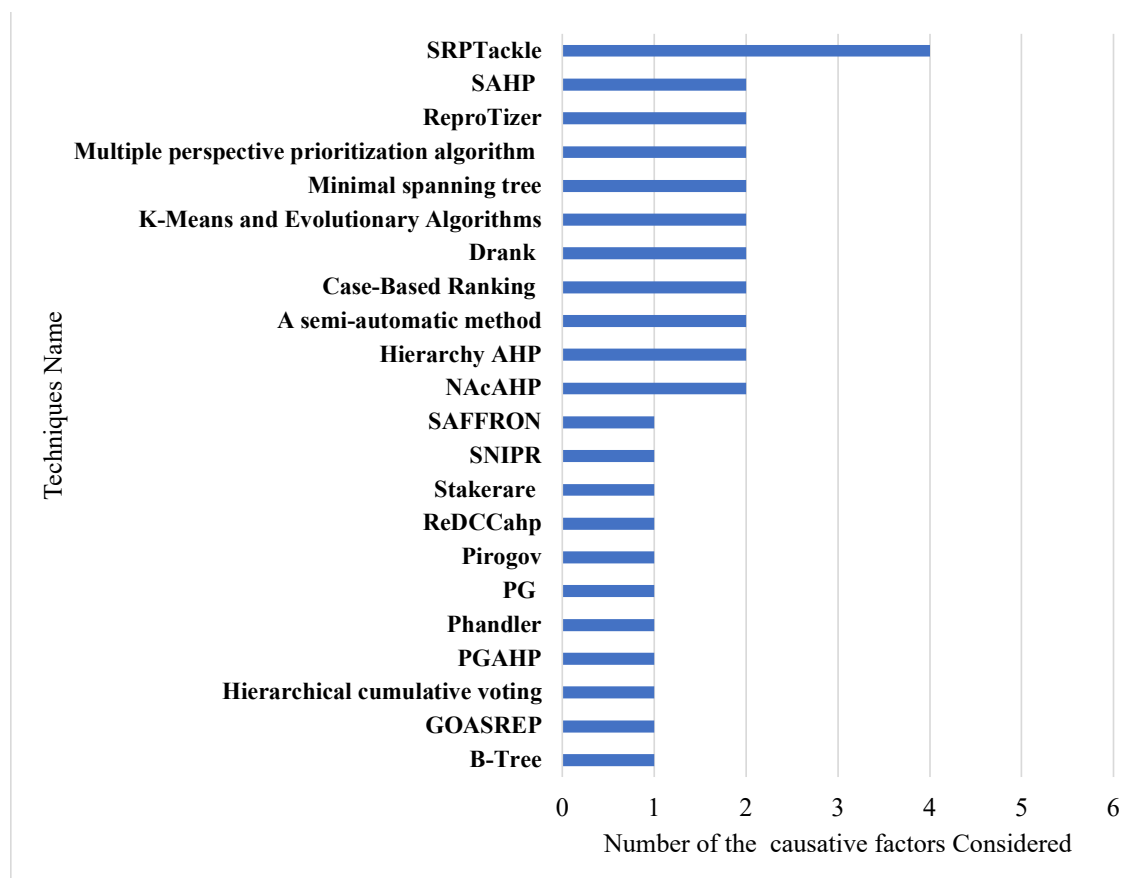


Figure 5.10: Number of the causative factors being covered in addressing the scalability issue by each identified technique.

Table 5.3: Analysis of evaluation extent for each identified techniques.

Name of the Techniques	Being evaluated	Evaluation Method	Name of the project and dataset used	Source of the project	Factor Considered
A semi-automatic technique [7]	Evaluated	Experiment	Historical database-geographic information systems (GIS) - 100 Requirements	Academic	Human efforts required, Scarcity of the automation
A generic framework [18]	Not evaluated	-	-	-	-
B-Tree [20]	Evaluated	Experiment	not mentioned - 121 requirements	Academic	Number of Comparisons
Case-Based Ranking [21]	Evaluated	Case Study	ProVotE -104 requirements	Industrial	Number of Comparisons, Human efforts required
Drank [22]	Evaluated	Experiment	Book Trading System (BTS) and Library Management System (LMS)	Industrial	Number comparisons, Scarcity of the automation
GOASREP [23]	Evaluated	Case Study	Institute Examination System	Industrial	Human efforts required
Hierarchy AHP [12]	Evaluated	Experiment	Telephony system	Industrial	Number of comparisons, Time

Table 5.3: Analysis of evaluation extent for each identified techniques (continued).

Name of the Techniques	Being evaluated	Evaluation Method	Name of the project and data set used	Source of the project	Factor Considered
Hierarchical cumulative voting [24]	Evaluated	Experiment	Course management system	Academic	Number of comparisons
K-Means and Evolutionary Technique [19]	Evaluated	Experiment	RALIC-Data set	Industrial	Number of Comparisons, Time
Minimal spanning tree [12]	Evaluated	Experiment	Telephony system	Industrial	Number of comparisons, Time
Multiple Perspective Prioritization [25]	Evaluated	Experiment	Online Flea Market System(OFMS)	Industrial	Time, Human efforts required
NAcAHP [5]	Evaluated	Experiment	Product manufacturing domain	Industrial	Time
PG [27]	Evaluated	Case Study	Personal Health Record (PHR) management system for Cancer patients	Industrial	Number of comparisons
PGAHP [26]	Evaluated	Experiment	Library Management System	Academic	Number of Comparisons
Phandler [9]	Evaluated	Experiment	Dataset 7 - 500 requirements	Industrial	Number of Comparisons

Table 5.3: Analysis of evaluation extent for each identified techniques (continued).

Name of the Techniques	Being evaluated	Evaluation Method	Name of the project and dataset used	Source of the project	Factor Considered
Pirogov [28]	Evaluated	Experiment	Ice Breaker System, and also on a set of stakeholders' raw feature requests mined from the discussion forum of an open source product named SugarCRM - 202 requirements	Industrial	Scarcity of the automation
ReproTizer [29]	Evaluated	Experiment	GSMS project (A web-based Graduate Students' information Management System in Universiti Teknologi Malaysia), Health information system (HIS) software-100 requirements, RALIC project-200 requirements, Enterprise resource planning (ERP) software package-500 and then, 1000 requirements	Academic and Industrial	Number of comparisons, Time
ReDCCahp [30]	Evaluated	Experiment	online flea market system OFMS (E-market application domain)	Industrial	Human efforts required
SAFFRON [33]	Evaluated	Case Study	RALIC dataset - 82 requirements	Industrial	Human efforts required
SAHP [34]	Evaluated	Case Study	On Demand Open Object (ODOO) ERP- 96 FRs	Industrial	Time, Human efforts required

Table 5.3: Analysis of evaluation extent for each identified techniques (continued).

Name of the Techniques	Being evaluated	Evaluation Method	Name of the project and dataset used	Source of the project	Factor Considered
SNIPR [32]	Evaluated	Experiment	-100 Requirements	Industrial	Time
SRPTackle [6]	Evaluated	Experiment	Ralic dataset- 122 requirements	Industrial	Time, Human efforts required, Number of comparisons, Scarcity of the automation
Stakerare [31]	Evaluated	Experiment	RALIC dataset- 112 requirements	Industrial	Human efforts required

5.7 RQ6. What are the limitations of the identified techniques in RQ3?

The aim of this question is to reveal the limitations of the identified technique RQ1 in solving the scalability issue. Through the identified limitations, we can get a better understanding of the potential improvement that can be introduced to improve the performance of these techniques in prioritizing the requirements in the context of scalability issues. To answer this question, each identified technique was analyzed in terms of the limitations that have been reported in the selected studies. The outcome of this analysis is presented in Table 5.4, in which the limitations of each identified technique are revealed, which can be explained as follows:

Requirements Interdependencies: As can be observed from Table 5.4, most of the existing techniques ignore the dependencies among the requirements during the prioritization process. Requirements dependency refers to the fact that the requirements are dependent on each other. This is an important attribute that determines the reliability of the preferred requirements. Interdependent requirements can ultimately be combined as equal which means that without one implementation others cannot be implemented. Thus, prioritizing these requirements without considering their dependencies can negatively affect the accuracy of the final prioritization results. Nonetheless, this attribute is rarely considered by the authors of requirement prioritization studies. Out of 23 identified techniques, only Drank technique [22] considered the dependencies among the requirements during the prioritization process by applying weighted PageRank. The evaluation results of this technique revealed the consideration of the requirements interdependencies assists to improve the accuracy of results improvement. However, this technique has the limitation of being heavily relied on the involvement of the expertise to perform the RP process [22].

Rank update: Rank update is another limitation considered in the existing technique. This is defined as the ability of the technique to automatically update the prioritization rank of the requirements at any time when adding or removing requirements from the list. This situation is related to the continued evolution of requirements. Out of 23 identified techniques, 2 techniques are only considered the rank update criteria: ReproTizer [29], SAFFRON [33]. Although these techniques are useful in terms of dealing with the rank update, these techniques ignore the requirements interdependencies and their RP process heavily relies on the need of the involvement of professional expertise. The rank update is an essential aspect to be addressed in RP as the decision-making and selection process cannot survive without repetition and the requirements of the most existing industrial projects keep changing. Therefore, an excellent and reliable prioritization technique is recommended to support the rank updates.

Heavily relying on the professional intervention: In the majority of the present techniques, heavily relying on the professional intervention is also seen as a constraint. To conduct the prioritization process, most of the techniques re-

quire the heavy involvement of experienced experts, such as Drank[22], hierarchical AHP[12],PGAHP[26], Phandler [9], Stakerare[31]. For this reason, understanding the requirements for the implementation of the methodology can be a complex problem with a lack of expertise. Also, the involvement of professional experts increases the probability of fault results due to the potential of bias induced by the judgment of the expertise which may affect the quality of the prioritization results [9], [15]. When assessing the relative impact levels of stakeholders participating in the requirement prioritization process or classifying requirements, expert bias might be detected, which may be affecting the validity of the technique in producing an accurate prioritization results [9], [15]. Although expert human assistance in domain knowledge and abilities may be critical in disclosing the prioritizing outcomes, human involvement in the prioritization process that is either unavailable or incorrect may skew the prioritization results [31].

Error proneness and the sensitivity to the judgemental errors:

Error proneness and the sensitivity to the judgemental errors in terms of producing accurate ranked list of requirement is another major limitation in the identified techniques. Error proneness and sensitivity to the error of judgement of the RP technique is related to the ability of the technique in producing accurate results that reflect actual prioritization results. The inability of producing an accurate prioritization results can be related to producing a lot of judgemental errors owing to the inability of catering the consistency. Techniques that embedded AHP in their prioritization processes like Hierarchy AHP, minimal spanning tree, and SAHP have an issue of producing unreliable accuracy results. For instance, Hierarchy AHP has succeeded in minimizing the number of comparisons by reducing the number of redundant comparisons. However, ignoring the redundancy of the pair-wise comparisons induce an inability to address consistency and raises the sensitivity to the judgemental errors as the existence of redundancy of the pairwise comparisons allows a consistency check where judgement errors can be revealed [14]. Similarly, the SAHP and minimal spanning tree are sensitive to judgment error, which produces inconsistent results and has weak fault tolerance [34]. Besides that, most of existing RP techniques have a complex methodology for prioritizing large set of requirements, and most of the existing are heavily relied on the professional expertise in doing the complex prioritization process, which induce to the threats that are linked to human biases and the validity of the technique in case there is a lack of expertise, which induce inability of technique in producing an accurate results.

Table 5.4: Analysis of the existing limitations of each identified selected studies.

Name of the Techniques	Limitations
A semi-automatic technique [7]	To be able to identify prioritization criteria from multiple available data sources using NLP and do not cater rank updates. The lack of RE repositories for training AI algorithms. The automatic association of newly identified requirements to the existing ones from the feature model.

Table 5.4: Analysis of the existing limitations of each identified selected studies (continued).

Name of the Techniques	Limitations
A generic framework [18]	Method is not evaluated yet and needs to be tested in the real environment settings, and ignores the requirements interdependencies.
B-Tree [20]	Need to be tested and validated in the collaborating environment settings, and does not cater to rank updates.
Case-Based Ranking [21]	It has the lack of ability to support the coordination with the stakeholders through negotiations.
Drank [22]	It requires the involvement of professional intervention for the RP process.
GOASREP [23]	Ignoring the dependencies among the requirements.
Hierarchy AHP [12]	Sensitive to the error of judgement and ignore the requirement interdependencies.
Hierarchical cumulative voting [24]	HCV does not consider the requirements dependency and does not cater to the rank updates.
K-Means and Evolutionary Technique [19]	Ignoring the dependencies among the requirements. Ignoring the ability of rank updates.
Minimal spanning tree [12]	It is sensitive to the error of judgement as all redundancies are eliminated.
Multiple perspective prioritization [25]	Does not consider requirement interdependencies and rank updates as well.
NAcAHP [5]	Does not consider requirement interdependencies.
PG [27]	It is ignoring the interdependencies among the requirements, does not reveal the priority value of each requirement, and does not cater to rank updates.
PGAHP [26]	The requirement prioritization process is dependent on the expert's involvement.
Phandler [9]	Require heavy involvement of highly professional business analysts and does not even consider the requirements interdependencies, does not consider rank updates.
Pirogov [28]	Error prone and does not return perfect precision or recall in the results.
ReproTizer [29]	It is ignoring requirements interdependencies.
ReDCCahp [30]	Does not fulfill the requirement interdependencies and ignores rank updates.
SAHP [34]	It is complex and time-consuming and sensitive to judgemental errors. do not consider rank updates.
SAFFRON [33]	It depends on expert participation and does not consider the interdependencies among the requirements.

Table 5.4: Analysis of the existing limitations of each identified selected studies (continued).

Name of the Techniques	Limitations
SNIPR [32]	Implementing the RP process requires extensive professional intervention.
SRPTackle [6]	It does not consider the requirements inter-dependencies while handling the large set of requirements, does not cater to rank updates.
Stakerare [31]	It does not consider requirement interdependencies, it heavily relies on the involvement of the professional experts, and does not cater to rank updates.

6

Threats to validity

Systematic literature review studies are often confronted with four types of validity threats (i.e., construct validity, conclusion validity, external validity and internal validity) [15].

Construct validity: One potential threat to the construct validity is concerning to completeness. We applied the established protocol with a rigorous review search technique. However, we cannot guarantee that we successfully captured all the related research studies as we just considered four digital libraries in the conducted searching strategy. Thus, there is no certainty that all relevant studies are included in this review. Furthermore, relevant materials written in languages other than English were not included, which means that essential or relevant studies may have been overlooked.

Conclusion validity: Another potential threat of this study is conclusion validity which refers to the bias in the selection of the final selected studies in this SLR. To minimize this threat, the study selection procedure was followed to the letter to ensure that each study included was relevant. To collect data from the selected primary studies, the data were also critically extracted and then qualitatively and quantitatively synthesized. In addition, a set of specific quality assessment criteria was used to avoid imprecise inclusion. However, we cannot ensure that the established inclusion and exclusion criteria, quality assessment criteria, and data synthesis are sufficient to prevent bias in the selection and synthesis of this review's relevant research.

External validity: Factors that impede the ability to generalize SLR findings outside the study boundaries is formed as a challenges to external validity. Non-English studies and grey studies were omitted, as stated in the indicated study selection process. The risks in this section determine if the final selected studies of this review are capable of constituting all forms of review studies in the RP realm. With the implementation of our designed review procedure, we could identified a typical group of studies that can reflect the domain knowledge from previous studies and come up with a comprehensive source of information for practitioners and researchers in the RP realm. Nevertheless, from an academic standpoint, our findings in this SLR are more concerning in the RP field that are from the online publication research sources than in industrial situations.

Internal validity: Another potential threat of this study is internal validity.

To measure the performance of the existing RP techniques in terms of scalability issue, this research only considered the RP studies that concerned on prioritizing the large set of requirements. There are some studies that are purposefully only used prioritization on small set of requirements due to limited resources such as there were not many case studies accessible at the time. Consequently, It's possible that their real performance on large projects wasn't adequately assessed. To minimize this issue we considered various kinds of publications: published journal papers, conference papers, workshops, and book chapter, and IEEE Bulletins.

7

Conclusion

7.1 Discussion

This systematic literature review provides a comprehensive investigation of the scalability issues in the area of requirement prioritization. Specifically, this SLR was conducted to critically analyze the reasons that lead the existing techniques to be not able to work with a large set of requirements by identifying and analyzing the factors that cause this problem and to what extent the existing techniques have catered these identified factors and addressed the scalability issue along with revealing the limitations of the identified techniques in solving the scalability issue and recommending future recommendations.

In RQ1, we revealed the factors that cause the scalability issue in the prioritization process, along with their descriptions and citations for existing studies that reported each factor. Four factors are revealed from the analysis of the final primary studies: number of comparisons, time, human efforts required, and scarcity of automation in conducting the prioritization process.

Furthermore, the results of RQ2 presented a detailed analysis of the identified factors to reveal the impact of each factor towards inducing the scalability issue and measure the significant degree of the listed factors in raising the scalability issue in the RP process. The significant degree was measured by the usage frequency for each listed factor from the primary research studies. We found that the number of comparisons is the most significant factor that can lead to the inability to deal with large-scale requirements in prioritization, with a percentage of 48%.

In RQ3 and RQ4 of this SLR, we also revealed the existing techniques that have been proposed to handle the scalability issue, along with analyzing these identified techniques distinctly in terms of their software development usage contexts, revealing features used by each identified technique in solving the scalability issue and their usage consequences. 23 techniques have been identified that have been proposed with the aim of addressing the scalability issue. We noticed that the identified RP techniques have been applied or used in 12 usage contexts. SRP context was revealed as the highest usage context in being applied by existing techniques with the percentage of 33% as nowadays most of software development organizations follow the SRP as a context for handling the large scale of data on the basis of release planning. The features used by the identified techniques and their us-

age consequences were revealed, and new categorization of the identified technique was introduced in terms of their features used: Machine learning based techniques (techniques that adapt any machine learning and AI algorithms), and combination of the existing based techniques (techniques that adapt combination of the existing methods for prioritizing large set of requirements).

Concerning the results of the RQ5, a detailed analysis was conducted for each identified technique in terms of being evaluated or not, the evaluation environment setting, revealing the causative factors being considered in addressing the scalability issue which is formatted. It can be observed that out of 23 identified techniques, 22 were evaluated, which refers to 96% percent, and only one technique was not evaluated as it was introduced in the form of a generic framework. Among the 22 evaluated techniques, 17 were evaluated using experiment study, and the rest (5 techniques) were evaluated using the case study field. In addition, we found that most of the identified techniques were evaluated within the industrial settings with a percentage of 82%, which reflects the serious intention of researchers towards addressing the scalability issue in real practices. Moreover, the ratio of considering each identified causative factor in the evaluation of the identified techniques was revealed in this SLR. Out of 22 evaluated techniques, the number of comparisons factor has been considered in the evaluation of 13, which were associated with the highest rank of considered factors in the evaluation of the identified techniques with respect to addressing the scalability issue. Moreover, we found that only one technique (SRPTackle) considered all four factors (human efforts required, time, number of comparisons, scarcity of the automation) during its performance evaluation. This finding reveals that most of the identified methods do not adequately consider the identified four causative factors in their evaluation performance and mostly cover up to two factors. Therefore, a comprehensive assessment of the performance of these techniques requires considering the four causative factors to ensure their ability to address scalability issues effectively.

Through the identified limitations in RQ6, we can get a better understanding of the potential improvement that can be introduced to improve the performance of these techniques in prioritizing large set requirements in the context of scalability issues. As can be observed that four significant limitations were identified, e.g., ignoring the requirements dependencies, error-prone and sensitivity to judgemental errors, rank update, heavily relying on professional intervention and most of the existing techniques ignore the dependencies among the requirements during the prioritization process. Thus, we call for new research to propose a new technique that can scale well with large requirements along with the ability to address these identified challenges and consider the four causative factors during the evaluation performance.

7.2 Future Recommendations

On the basis of the conducted analysis of the limitation of the existing RP techniques that have been proposed to address the scalability, these techniques still have

some major limitations. Ignoring the requirements dependencies, error-prone and sensitivity to judgemental errors, rank update, and heavily relying on professional intervention are the common limitations of the existing RP techniques as identified in Table 5.4 and further described in the RQ6. Hence, some significant opportunities are addressed below for future research on improving the existing RP techniques.

Catering the requirements dependencies among the requirements during the prioritization process. Requirement interdependencies are considered as an important attribute that determines the reliability of the requirements. Interdependent requirements can ultimately be considered before ranking the requirement. For instance, a dependency value for each individual requirement with others should be calculated with the involvement of the system development team, who can assist in revealing the dependencies among the requirements; then, the prioritization process should be conducted with considering the obtained dependency value of each requirement. Considering the requirements dependencies in the RP process will assist the accuracy of the final prioritization results.

Rank Update refers to the ability of the technique to automatically update the prioritization rank of the requirements at any time when adding or removing requirements from the list. From this research, it can be noticed that the rank update is an essential aspect to be addressed in prioritizing the large set of requirements to make the decision and selection process reliable as the requirements of most existing industrial projects keep changing. Automating the process and using collaborative filtering approaches can assist in automating the ranking process when updating the priority list after prompting new requirements that arise in large-scale projects.

As it has been observed that most of the techniques have limitations of rank update, error-prone and the sensitivity to judgemental errors, and require the heavy involvement of experienced experts. This can be addressed by automating the prioritization process by employing scalable and rigorous AI techniques with big data-related algorithms that can deal with large amounts of data. Such a solution can assist in reducing the number of the comparisons needed without being error-prone and the sensitivity to judgemental errors, and reducing heavy reliance on professional intervention in the prioritization process, specifically in assigning the priority value for each requirement, which can also reduce the judgemental errors that raised by the potential bias induced by the judgment of the expertise.

7.3 Conclusion

This thesis paper is gives a wide insightful review of the scalability issue in the terms of requirement prioritization domain by identifying the causative factors and how their cause in the scalability in RP, the existing techniques that have been proposed to handle the scalability issue, the features used by the techniques in solving the scalability issue and revealing their usage consequences and the revealed the challenges of the RP techniques which introduce to address the scalability issue.

In this thesis, the author designed a review protocol that was based on the guidelines of SLR proposed by Kitchenham [17]. The research questions of this SLR were formulated on the basis of the stated objective. In this SLR authors collect the potential related existing works in two stages. Initially, 258 potential related studies have been compiled, then after rigorous selection procedure, 22 studies have been finally selected as the primary studies.

The results of the study revealed four factors that raise the scalability issue in the prioritisation process : number of comparisons, time, human efforts required and scarcity of the automation. Analysis of these identified causative factors have been conducted to reveal how the identified factors induce the scalability issue in prioritizing the large scale requirements. Twenty-three techniques that proposed to handle the scalability issue have been identified and analysed in terms of their software development usage contexts, features used by the each technique in solving the scalability issue and revealing their usage consequences, and the evaluation extent of identified RP techniques in addressing the identified causative factors, and limitations. The findings revealed that these techniques still face certain major limitations with respect to ignore the requirement dependencies, error-prone and the sensitivity to the judgemental errors, rank update, involvement of experienced experts.

To conclude, these SLR challenges are revealed from the existing techniques named requirement dependencies, error-prone, judgmental error, rank update and involvement of experienced expert. There is a solid ought to receive the distinguished current qualities and address their inborn confinements. Hence, the current work presents a little jump forward that can serve as guidelines for future advancement of RP techniques.

References

- [1] F. A. Bukhsh, Z. A. Bukhsh, and M. Daneva, “A systematic literature review on requirement prioritization techniques and their empirical evaluation,” *Computer Standards Interfaces*, vol. 69, p. 103 389, 2020.
- [2] T. Kamal, Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei, and A. Alsanad, “Identification and prioritization of agile requirements change management success factors in the domain of global software development,” *IEEE Access*, vol. 8, pp. 44 714–44 726, 2020.
- [3] S. Ali, Y. Hafeez, S. Hussain, S. Yang, and M. Jamal, “Requirement prioritization framework using case-based reasoning: A mining-based approach,” *Expert Systems*, vol. 38, 8 2021, cited By 0.
- [4] S. Malgaonkar, S. A. Licorish, and B. T. R. Savarimuthu, “Understanding requirements prioritisation: Literature survey and critical evaluation,” *IET Software*, vol. 14, pp. 607–622, 6 Dec. 2020, cited By 0.
- [5] S. Nidhra, L. P. Kelapanda Satish, and V. S. Ethiraj, “Analytical hierarchy process issues and mitigation strategy for large number of requirements,” in *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, 2012, pp. 1–8.
- [6] F. Hujainah, R. B. A. Bakar, A. B. Nasser, B. Al-haimi, and K. Z. Zamli, “Srp-tackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects,” *Information and Software Technology*, vol. 131, p. 106 501, Mar. 2021, cited By 0.
- [7] M. I. L. Lunarejo, “Requirements prioritization based on multiple criteria using artificial intelligence techniques,” *IEEE*, Sep. 2021, pp. 480–485.
- [8] H. Zhang, M. Zhang, T. Yue, S. Ali, and Y. Li, “Uncertainty-wise requirements prioritization with search,” *ACM Transactions on Software Engineering and Methodology*, vol. 30, pp. 1–54, 1 Jan. 2021, Included.
- [9] M. I. Babar, M. Ghazali, D. N. Jawawi, S. M. Shamsuddin, and N. Ibrahim, “Phandler: An expert system for a scalable software requirements prioritization process,” *Knowledge-Based Systems*, vol. 84, pp. 179–202, 2015.
- [10] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, “A systematic literature review of software requirements prioritization research,” *Information and Software Technology*, vol. 56, pp. 568–585, 6 2014.
- [11] P. Berander and P. Jönsson, “Hierarchical cumulative voting (hcv) prioritization of requirements in hierarchies,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, pp. 819–849, 6 2006.

-
- [12] J. Karlsson, C. Wohlin, and B. Regnell, “An evaluation of methods for prioritizing software requirements,” *Information and Software Technology*, vol. 39, pp. 939–947, 1998.
- [13] Hujainah, Bakar, Al-Haimi, and Nasser, “Analyzing requirement prioritization techniques based on the used aspects,” *Res. J. Appl. Sci.*, 2016.
- [14] P. Berander and A. Andrews, “Requirements prioritization,” in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 69–94.
- [15] F. Hujainah, R. B. A. Bakar, M. A. Abdulgaber, and K. Z. Zamli, “Software requirements prioritisation: A systematic literature review on significance, stakeholders, techniques and challenges,” *IEEE Access*, vol. 6, pp. 71 497–71 523, 2018.
- [16] F. Hujainah, R. B. A. Bakar, and M. A. Abdulgaber, “Investigation of requirements interdependencies in existing techniques of requirements prioritization,” *Tehnicki vjesnik - Technical Gazette*, vol. 26, pp. 1186–1190, 4 Aug. 2019, cited By 6.
- [17] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” vol. 2, Jan. 2007.
- [18] K. AbdElazim, R. Moawad, and E. Elfakharany, “A framework for requirements prioritization process in agile software development,” *Journal of Physics: Conference Series*, vol. 1454, p. 012 001, 1 Feb. 2020, cited By 2.
- [19] P. Achimugu and A. Selamat, *Computational Intelligence Applications in Modeling and Control*, A. T. Azar and S. Vaidyanathan, Eds. Springer International Publishing, 2015, vol. 575, pp. 73–93.
- [20] R. Beg, Q. Abbas, and R. P. Verma, “An approach for requirement prioritization using b-tree,” *IEEE*, 2008, pp. 1216–1221.
- [21] A. Perini, A. Susi, and P. Avesani, “A machine learning approach to software requirements prioritization,” *IEEE Transactions on Software Engineering*, vol. 39, pp. 445–461, 4 2013.
- [22] F. Shao, R. Peng, H. Lai, and B. Wang, “Drank: A semi-automated requirements prioritization method based on preferences and dependencies,” *Journal of Systems and Software*, vol. 126, pp. 141–156, 2016.
- [23] S. Chandra, S. Vikrant, B. Siba, K. U. Prasant, and K. Pattnaik, “Goasrep: Goal oriented approach for software requirements elicitation and prioritization using analytic hierarchy,” vol. 516, 2017.
- [24] P. Berander and M. Svahnberg, “Evaluating two ways of calculating priorities in requirements hierarchies - an experiment on hierarchical cumulative voting,” *Journal of Systems and Software*, vol. 82, pp. 836–850, 5 2009.
- [25] I. Ibriwesh, S.-B. Ho, I. Chai, and C.-H. Tan, “Prioritizing solution-oriented software requirements using the multiple perspective prioritization technique algorithm: An empirical investigation,” *Concurrent Engineering*, vol. 27, pp. 68–79, 1 Mar. 2019, cited By 5.
- [26] K. Ayub, F. Azam, M. W. Anwar, A. Amjad, and M. S. Jahan, “A novel approach for software requirement prioritization based upon non functional requirements,” *IEEE*, Oct. 2019, pp. 8–15.

-
- [27] H. Tufail, I. Qasim, M. F. Masood, S. Tanvir, and W. H. Butt, "Towards the selection of optimum requirements prioritization technique: A comparative analysis," *IEEE*, Mar. 2019, pp. 227–231.
- [28] C. Duan, P. Laurent, J. Cleland-Huang, and C. Kwiatkowski, "Towards automated requirements prioritization and triage," *Requirements Engineering*, vol. 14, pp. 73–89, 2 Jun. 2009.
- [29] P. Achimugu, A. Selamat, and R. Ibrahim, "Reprotizer: A fully implemented software requirements prioritization tool," N. T. Nguyen and R. Kowalczyk, Eds., Springer Berlin Heidelberg, 2016, pp. 80–105.
- [30] I. Ibriwesh, S.-B. Ho, and I. Chai, "Overcoming scalability issues in analytic hierarchy process with redcahp: An empirical investigation," *Arabian Journal for Science and Engineering*, pp. 1–17, Apr. 2018.
- [31] S. L. Lim and A. Finkelstein, "Stakerare : Using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE Transactions on Software Engineering*, vol. 38, pp. 707–735, 3 2012.
- [32] J. McZara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment," *Empirical Software Engineering*, vol. 20, pp. 1721–1761, 6 Dec. 2015, time consuming,
- [33] S. A. Asif, Z. Masud, R. Easmin, and A. U. Gias, "Saffron : A semi-automated framework for software requirements prioritization," *International Journal of Advanced Computer Science and Applications*, vol. 8, pp. 491–499, 12 2017.
- [34] M. Yaseen, A. Mustapha, and N. Ibrahim, "Prioritization of software functional requirements from developers perspective," *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 210–224, 9 2020, cited By 0.
- [35] P. Avesani, C. Bazzanella, A. Perini, and A. Susi, "Facing scalability issues in requirements prioritization with machine learning techniques," 2005, pp. 297–305.
- [36] M. I. Babar, M. Ramzan, and S. A. K. Ghayyur, "Challenges and future trends in software requirements prioritization," *IEEE*, Jul. 2011, pp. 319–324.
- [37] M. S. Hasan, A. A. Mahmood, J. Alam, and S. N. Hasan, "An evaluation of software requirement prioritization techniques," *International Journal of Computer Science and Information Security*, vol. 8, pp. 83–94, 9 2010.
- [38] M. A. Iqbal, A. M. Zaidi, and S. Murtaza, "A new requirement prioritization model for market driven products using analytical hierarchical process," 2010, pp. 142–149.
- [39] K. A. Khan, "A systematic review of software requirements prioritization," Master's Thesis, Blekinge Institute of Technology, Ronneby, Sweden, 2006.
- [40] Q. Ma, "The effectiveness of requirements prioritization techniques for a medium to large number of requirements : A systematic literature review," Master's Thesis, School of Computing and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand, 2009, p. 92.
- [41] N. Kukreja, B. Boehm, S. S. Payyavula, and S. Padmanabhuni, "Selecting an appropriate framework for value-based requirements prioritization," 2012, pp. 303–308.

- [42] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, “Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritisation techniques,” *Empirical Software Engineering*, vol. 12, pp. 3–33, 1 Jan. 2007.
- [43] P. Tonella, A. Susi, and F. Palma, “Using interactive ga for requirements prioritization,” *IEEE*, Sep. 2010, pp. 57–66.
- [44] M. Ramzan, M. A. Jaffar, M. A. Iqbal, S. Anwar, and A. a. Shahid, “Value based fuzzy requirement prioritization and its evaluation framework,” *2009 4th International Conference on Innovative Computing, Information and Control, ICICIC 2009*, pp. 1464–1468, 2009.
- [45] M. Yaseen, A. Mustapha, and N. Ibrahim, “An approach for managing large-sized software requirements during prioritization,” 2018, pp. 98–103.
- [46] C. Wohlin, M. Höst, and K. Henningsson, *Empirical research methods in software engineering*, 2003.

A

Appendix 1