IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 3, 2018, accepted November 7, 2018, date of publication November 19, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881755

Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges

FADHL HUJAINAH[1], ROHANI BINTI ABU BAKAR[1], MANSOOR ABDULLATEEF ABDULGABBER[2], AND KAMAL Z. ZAMLI[1], (Member, IEEE)

[1]Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan 26300, Malaysia
[2]Faculty of Computer and Cyber Science, Software Engineering, University Prince Mugrin, Al-Madinah 41499, Saudi Arabia

Corresponding author: Fadhl Hujainah (fadelhogina@gmail.com)

This work was partially supported by the Fundamental Research Grant: A New Model for Automated Stakeholder Quantification and Prioritization Based on User Needs for Software System Development from the Ministry of Higher Education Malaysia under Grant RDU190164 and the PGRS170393 Grant from Universiti Malaysia Pahang.

ABSTRACT As one of the gatekeepers of quality software systems, requirements' prioritization (RP) is often used to select the most important requirements as perceived by system stakeholders. To date, many RP techniques that adopt various approaches have been proposed in the literature. To identify the strengths, opportunities, and limitations of these existing approaches, this paper studied and analyzed the RP field in terms of its significance in the software development process based on the standard review guidelines by Kitchenham. By a rigorous study selection strategy, 122 relevant studies were selected to address the defined research questions. Findings indicated that RP plays a vital role in ensuring the development of a quality system with defined constraints. The stakeholders involved in RP were reported, and new categories of the participating stakeholders were proposed. Additionally, 108 RP techniques were identified and analyzed with respect to their benefits, prioritization criteria, size of requirements, types in terms of automation level, and their limitations; 84 prioritization criteria were disclosed with their frequency usages in prioritizing the requirements. The study revealed that the existing techniques suffer from serious limitations in terms of scalability, the lack of quantification, and the prioritization of the participating stakeholders, time consumption, requirement interdependences, and the need for highly professional human intervention. These findings are useful for researchers and practitioners in improving the current state of the art and state of practices.

INDEX TERMS Requirements prioritization, stakeholders, techniques, challenges, systematic literature review, requirements prioritization criteria.

I. INTRODUCTION

Requirement engineering (RE) is one of the most essential phases in software development. RE is mainly concerned with the process of eliciting, documenting and maintaining stakeholders' requirements [1], [2]. Often, meeting and securing stakeholders' core requirements is one of the main reasons for producing a good-quality software system [3]–[5].

One important aspect of RE is requirements prioritization (RP). As the name suggests, RP relates to the process of identifying the most essential requirements for the implementation of a successful system [5], [6]. RP is an iterative process [7] that involves critical and complex decision-making activities that facilitate the development of a high-quality system within defined constraints [5], [8]. Specifically, RP ensures the correct ordering of requirements' implementation as perceived by stakeholders [9]–[13] (i.e. by rearranging the requirements according to importance using various prioritization criteria, such as importance, cost, penalty and risk [6], [10], [14], [15]). Here, the stakeholders' involvements often lead to an accurate prioritization result.

Thus, the selection process of stakeholders who are involved in the RP process is crucial in the RP domain.

In line with the ever-increasing demands for software functionalities, most recent system projects include many requirements. As such, implementing all of the requirements with limited resources (e.g. insufficient budget, time and technical staff [12], [16], [17]) is extremely difficult. Therefore, development teams tend to deliver system requirements to stakeholders in stages (i.e. with a number of small releases); each release contains an incremental number of requirements from all the extracted requirements. Selecting the important requirements to be developed and delivered first in the early releases is important in meeting stakeholders' demands [16], [18]–[21]. The requirements that are less essential are left for latter releases. Any insufficient software development resource is neglected or delayed [20], thereby optimizing resource usage.

Nonetheless, RP is a challenging task. Different features of software requirements must be considered in prioritizing requirements, such as dependency [8], cost–value [22] and other features. Many useful techniques have been successfully developed to execute the RP process, including PHandler [5], StakeRare [4] and DRank [8].

To investigate the strengths and limitations of existing RP techniques, many review studies have been conducted (e.g. Khan [23], Kaur and Bawa [24], Pergher and Rossi [25], Pitangueira *et al.* [7], and Achimugu *et al.* [26]). At a glance, these review studies have usefully emphasized the performance of existing RP techniques. Nonetheless, a close look reveals two main limitations. Firstly, these existing review studies have not sufficiently focused on the analysis of RP in terms of the characteristics of decision makers, the prioritization criteria they use, RP activity in the software development context and significance of RP in the software development process.

Secondly, given that new RP techniques have been introduced in the literature, an up-to-date analysis of existing work is needed. Such analysis is helpful for researchers and practitioners in improving the current state of the art and state of practices. To date, the most recent review is from Achimugu *et al.* [26], in which 49 RP techniques were analyzed. Unlike Achimugu et al., we incorporated the stakeholders' dimension as a new evaluation criteria apart from covering additional RP techniques (i.e. 108). Thus, the contribution of this research can be summarized as follows:

- Analysis and review of the characteristics of participating stakeholders in RP
- A new perspective on the RP activity within software development contexts and the significance of RP in the software development process and
- Empirical evidence for uncovered and recent RP techniques and their limitations.

The remainder of this paper is structured into five sections. Section 2 reports the related works. Section 3 describes the review methodology that was used to conduct this review. Section 4 elaborates the research results and discussions. Section 5 presents a detailed explanation of threats to the validity of this research. Finally, Section 6 concludes this study.

II. RELATED WORK

Five review studies related to the RP process were collected.

Khan's [23] was perhaps the earliest systematic literature review (SLR) on software RP. This review aimed to objectively compare RP techniques from eight selected studies. The authors concluded that most of the proposed RP techniques addressed only a small set of requirements.

Complementing the work of Kaur *et al.* [24] focused on studying and comparing the performance of seven RP techniques that are based on cumulative voting, analytic hierarchy process (AHP), numerical assignment, value-oriented prioritization, binary search tree, planning game and B-tree prioritizations. The performance of the selected techniques was evaluated on the basis of the criteria of measurement scale, time consumption, granularity, complexity and fault tolerance. The authors concluded that the area of RP still required additional work to enhance the effectiveness of RP techniques in terms of complexity, fault tolerance and time consumption.

Pergher and Rossi [25] presented a systematic mapping study in software RP to highlight the RP area that had been explored by existing research studies and to clarify the state of the art in the conducted empirical research in RP. The review revealed that most of the existing studies mainly concentrated on techniques, whereas the existing empirical research was concerned with techniques and the issue of accuracy in RP.

Pitangueira *et al.* [7] presented an SLR on RP with specific focus on search-based software engineering (SBSE). The objective of the review was to investigate, categorize, analyze and classify the SBSE techniques that had been introduced to solve the issues of software RP and selection. Thirty-nine (39) relevant studies were selected and analyzed after executing the defined study selection process of the review. The review presented the requirements selection aspects, prioritizations issues and the proposed search techniques to address the specified issues.

Recently, Achimugu *et al.* [26] conducted an SLR of RP techniques; the review focused on measurement scales, descriptions and limitations. The findings suggested that the existing techniques still faced a number of challenges related to time consumption, requirement interdependencies and scalability.

Table 1 summarizes the findings (in terms of similarities and differences) of the related studies. The table indicates that the main concern of most of the related studies was to provide an overview of RP techniques' performance. With the exception of Achimugu *et al.* [26], which covered 49 RP techniques, most existing studies focused on only a few RP techniques. Although useful, the work of Achimugu et al falls short in terms of not giving sufficient consideration for the stakeholders' dimension within RP contexts. Additionally, recent developments in terms of newly

TABLE 1. Summary of related studies.

Study Reference	Covered Findings	Findings Similar to those of this SLR	Uncovered Findings Added into this SLR
Khan [23]	Highlighted detailed overview of eight RP techniques from eight selected studies.	Overview of RP. Empirical evidence for the eight compared RP techniques.	Detailed and specific overview of the significance of the RP process in software development. Detailed investigation of the usage prioritisation criteria in RP process. Empirical evidence for other RP techniques. Participating stakeholders in RP process. Detailed analysis of the limitations and benefits, dataset to be prioritised and type of each RP technique. Detailed analysis of the usage contexts of RP techniques and selected studies
Kaur and Bawa [24]	Conducted performance overview of seven common RP techniques with respect to their measurement scale, time consumption, granularity, complexity and fault tolerance.	Empirical evidence of the seven selected RP techniques.	Overall overview of RP. Detailed and specific overview of the significance of the RP process in software development. Detailed investigation of the usage prioritisation criteria in the RP process. Empirical evidence of other RP techniques. Participating stakeholders in the RP process. Detailed analysis of the limitations and benefits, dataset to be prioritised and type of each RP technique. Detailed analysis of the usage contexts of RP techniques and selected studies.
Pergher and Rossi [25]	Investigated RP areas that had been explored by existing studies and clarified the state of the art in the conducted empirical research in RP.	Overall overview of RP.	Detailed and specific overview of the significance of the RP process in software development. Detailed investigation of the usage prioritisation criteria in RP. Empirical evidence for other RP techniques. Participating stakeholders in RP. Detailed analysis of the limitations and benefits, dataset to be prioritised and type of each RP technique. Detailed analysis of usage contexts of RP techniques and selected studies.
Pitangueira et al. [7]	Analysed the SBSE approaches that had been introduced by the researcher to solve issues of software RP.	Overall overview of RP.	Detailed and specific overview of the significance of RP in software development. Detailed investigation of the usage prioritisation criteria in RP. Empirical evidence of other RP techniques. Analysis of the participating stakeholders in RP. Detailed analysis of the limitations and benefits, size of the prioritised requirements set and type of each RP technique. Detailed analysis of usage contexts of RP techniques and selected studies.
Achimugu et al. [26]	Analysed steps involved in the prioritisation process. Analysed 49 RP techniques in terms of limitations, description and measurement scale.	Overall overview of RP. Detailed overview of the 49 RP techniques' limitations.	Detailed and specific overview of the significance of the RP process in software development. Detailed investigation of the usage prioritisation criteria in the RP process. Analysis of the participating stakeholders in RP. Empirical evidence of other uncovered and recent RP techniques and their limitations. Detailed analysis of and benefits, usage size of the requirements set to be prioritised and type of each RP technique. Detailed analysis of usage contexts of RP techniques and selected studies.

developed RP techniques have also not been covered. Our SLR aimed to close this gap and address these issues by considering a large number of RP techniques.

The present study not only focuses on an overview of RP techniques but also investigates the characteristics of decision makers and the prioritizations criteria they use.

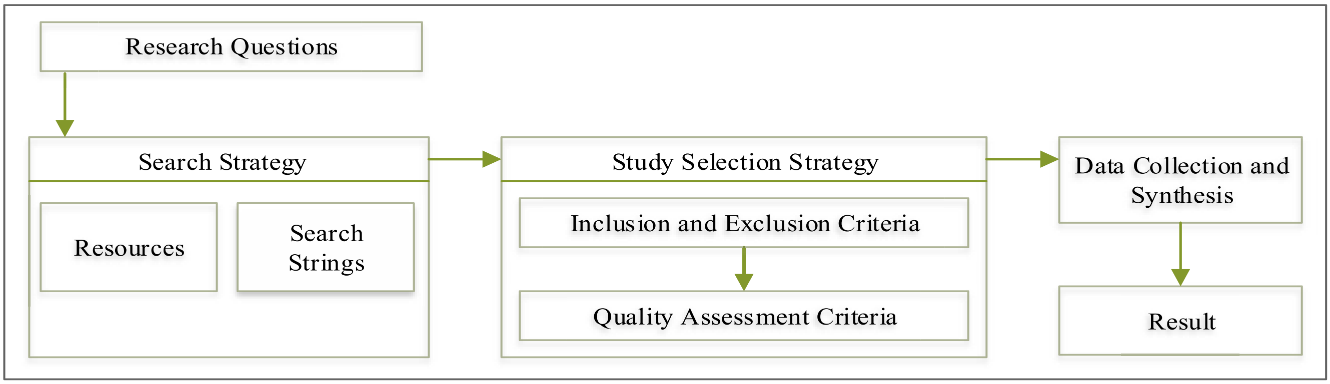


FIGURE 1. Review protocol.

Complementing existing work, this SLR research also analyzed the important impacts of implementing RP on system development, particularly emphasizing the extent to which the execution of RP can generate high-quality systems in real scenarios.

III. RESEARCH METHODOLOGY

To conduct this study, a review protocol was designed on the basis of the standard SLR guidelines of Kitchenham and Charters [27]. Figure 1 illustrates the review protocol of this work. The adopted research review protocol includes four activities: development of research questions, implementations of search and study selection strategies (inclusion and exclusion criteria, and quality assessment criteria [QAC]) and data collection along with data synthesis.

The following sub-sections explain the defined activities that were conducted to obtain the results of this study.

A. RESEARCH QUESTIONS

The aim of this study was to study, analyze and summarize the RP domain in terms of its significance in the system development process, existing RP techniques, stakeholders involved in the prioritization process, RP challenges or limitations and future sets for further research. To achieve this aim, the following research questions were defined.

- RQ1: What is the significance of conducting RP in the software development process?
- RQ2: What current techniques are available in executing RP and their prioritization criteria, types, benefits, size of requirements to be prioritized and limitations?
- RQ3: Who are the stakeholders involved in RP, and how can we classify them?
- RQ4: What are the usage contexts of the identified RP techniques in RQ2 and the selected studies?
- RQ5: What are the recommended enhancements for the specified challenges or limitations?

By answering these defined questions, we aimed to identity and emphasize the significance of conducting RP in system development; that is, we intended to provide a clear understanding and discussion regarding the reasons of prioritizing requirements and how they can help in producing high-quality systems within real industrial practices. Additionally, we aimed to report and classify the stakeholders that should participate in the RP process. Often, conducting RP with the participation of inadequate stakeholders affects the result of obtaining an accurate prioritized list of requirements with the possibility of missing the core requirements, thereby promoting system failure [28]–[30].

In addition, we intended to collect information about the available techniques that can be used to perform RP and analyze them with respect to their limitations, prioritization criteria, types, benefits, size and sets of requirements to be prioritized. The contexts of the RP techniques and the selected studies in this work that are proposed or applied in the system development context were also revealed. This analysis will help provide a clear picture of each technique, thereby assisting academic researchers and industrial requirement engineers in terms of selecting suitable techniques for adoption in prioritizing requirements in the system development process. Furthermore, the recommended future sets were presented to assist and encourage further research.

B. SEARCH STRATEGY

To comprehensively search for related studies, our search strategy began with an online search of digital libraries. Studies that are related to this review were extracted from seven main electronic database resources, namely, ScienceDirect, IEEE Xplore, Springer, ACM Digital Library, ISI Web of Science, Google Scholar and Scopus.

These digital libraries were chosen because they are considered relevant search engines for SLRs in software engineering [31]. Furthermore, they provide options of conducting advanced search by keywords and result filtering by publication type and year or by domain area.

Related studies were collected from various online publications, such as published conference proceedings, journal papers, book chapters, symposia, IEEE bulletins and workshops. However, specifying keywords or search terms is essential in performing online searches of electronic databases [27]. To ensure review quality, search terms were formulated on the basis of the defined research questions along with a stepwise procedure, which is as follows:

1 Identifying the main terms on the basis of the respective research questions
2 Finding the alternative synonyms and spelling of the main terms
3 Verifying the search terms of relevant studies and
4 Using the Boolean OR/AND operators to combine the search terms.

A list of search strings was identified to search for relevant studies. The output search strings that were used in this study are as follows:

- Requirements prioritization (OR / AND) selection.
- Requirement Prioritization in OR for Software Release Planning OR Release Planning.
- Significance (OR / AND) Importance (OR / AND) impacts of requirements prioritization.
- Requirements prioritization techniques OR methods OR frameworks, OR approaches.
- Requirements prioritization AND stakeholders.
- Stakeholders' roles (OR / AND) types of the requirements prioritization.
- Stakeholders in the requirements prioritization.
- Limitations OR challenges OR issues of requirement prioritization techniques.
- Context (OR / AND) domain (OR / AND) principle of requirements prioritization OR RP activity within software development context (OR / AND) domain (OR / AND) principle.
- Benefits OR advantages of requirement prioritization techniques OR methods OR frameworks, OR approaches.
- Size of requirements OR size set of requirements in requirements prioritization OR techniques OR methods OR frameworks, OR approaches.
- Criteria OR aspects OR attributes of requirement prioritization techniques OR methods OR frameworks, OR approaches.

All search strings were combined with Boolean operators (AND, OR) to extend the searching of studies and to increase the relevance of the search process. We implemented the search strings to the titles, abstracts and keywords of the papers in the identified electronic databases and then retrieved papers that include one of the identified search strings.

C. STUDY SELECTION STRATEGY

In the initial stage of the search process, 878 prospective studies were collected from the online digital libraries. To produce accurate and precise answers for the specified research questions, we needed to critically evaluate and scrutinize each collected work. Thus, the study selection process was designed to conduct scrutiny.

Figure 2 presents the process of the study selection strategy that we used, which consists of two main phases: inclusion and exclusion criteria and QAC. The inclusion and exclusion criteria were formulated on the basis of the specified research questions. Complementing Fig. 2, Table 2 presents the formulated inclusion and exclusion criteria of this review. The titles, abstracts and content of each collected study were concisely studied. Thus, studies that cannot provide potential answers to the listed research questions were excluded. We included published studies that are written in English and excluded those published in other languages.

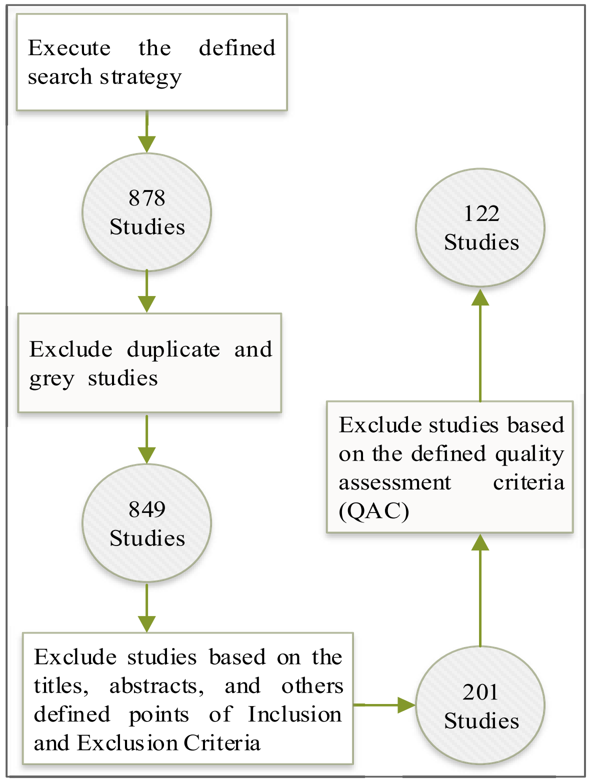


FIGURE 2. Study selection process.

TABLE 2. Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
• All studies published in English. • Studies that can provide potential answers to research questions on the basis of content, keywords, titles and abstracts of the studies. • Empirical studies and experience reports based on experts.	• Studies that are not written and published in English. • Duplicate studies. We included the most recent and complete versions and excluded other copies of the same study. • Studies that are irrelevant to the research questions. • Grey studies.

Furthermore, for each study that has various versions, the recent and most complete one was included, and the other copies of the same study were excluded. Grey studies (works in progress, unpublished or non-peer-reviewed publications, such as studies published on websites or those that do not have such bibliographic details as publication date or type) were also excluded. Thus, 201 relevant studies were selected after the implementation of the defined inclusion and exclusion criteria. QAC were used to identify the most relevant search studies to the research domain. We formulated several QAC checklists that are based on the guidelines of Kitchenham and Charters [27] and Kitchenham *et al.* [32] with respect to the defined aim of this study. Table 3 presents the formulated quality checklist used to evaluate the quality of the 201 studies.

TABLE 3. Quality Assessment Criteri.

ID	Question	Answers' Point Scores
QAC1	Is the aim of the research sufficiently illustrated?	Yes = 2/ moderately = 1/ no = 0
QAC2	Is the context of research expounded well?	Yes = 2/ moderately = 1/ no = 0
QAC3	Does the study concentrate on the related domain of RQs?	Yes = 2/ moderately = 1/ no = 0
QAC4	Is the proposed technique/solution clearly explained?	Yes = 2/ moderately = 1/ no = 0
QAC5	Is the evaluation of proposed technique performed on adequate case studies or project data sets?	Yes = 2 / moderately = 1/ no = 0
QAC6	Is the result of the study clearly stated?	Yes = 2 / moderately = 1/ no = 0

The defined QAC checklist comprises six questions. The answer to each question can be 'Yes', 'Moderately' or 'No', which were assigned point scores of 2, 1 and 0, respectively. QAC was applied by a precise study of the titles, abstracts and content of each study, assignment of a quality score to every study and then calculation of the overall quality scores by the summation of all answer scores to the defined checklist questions.

Result comparisons and discussions were conducted by the authors to address contradictions and thus achieve a consensus. To ensure dependability of the findings, studies that obtained quality scores of less than 6 (which are less than half of the full quality score of 12) were excluded. Thus, 122 works were selected as primary studies of this review. Table 5 in Appendix A presents the selected studies with their corresponding reference numbers and final quality scores.

D. DATA COLLECTION AND SYNTHESIS

In this study, the data collection and referencing process was executed with use of the software EndNote. Furthermore, data were collected on the basis of the defined research questions. Each selected study was carefully analyzed to obtain any relevant data that can help in addressing the questions. Then, in the data synthesis step, proofs were collected from the data gathered from the selected studies to answer the stated research questions [27]. In this systematic review, the data were synthesized qualitatively and quantitatively. Data related to the significance of the RP process were analyzed critically to answer RQ1 by presenting the impact of the RP process on ensuring the success of the system development process.

To answer RQ2, techniques of RP are highlighted and visualized using a descriptive diagram in Fig. 6, in which the collected existing techniques from the selected studies are categorized according to execution type (manual, semi-automated and fully automated) used in prioritization. Prioritization criteria, benefits, size of requirements and limitations of each existing RP technique are reported. Each RP technique was critically analyzed, and the results are in Table 6 of Appendix B. The stakeholders involved in the prioritization process are reported in Table 4, and the categorization of the identified stakeholders is shown in Fig. 9 to provide answers to RQ3. To answer RQ4, the RP contexts of the listed techniques in RQ2 and the selected studies were identified; furthermore, the usage frequency of each identified context in the RP techniques is reported and visualized as a bar chart. The categorization of the selected studies based on their publication years and focus with respect to their contexts is illustrated as a scatter diagram. Finally, the recommended future trends are presented with descriptive features to answer RQ5.

IV. RESULTS AND DISCUSSION

The results and discussion of this SLR and an overview of the included studies are illustrated in this section. A total of 121 works were finally considered primary studies for this review. These primary studies consisted of 48 conference papers, 48 journal papers, 11 book chapters, 6 published theses, 5 workshop papers and 2 papers for IEEE bulletins and symposia. Figure 3 illustrates the percentage for each publication channel of the selected primary studies. The figure shows that the conference and journal publication type have the highest percentage of 40%. The type of book chapters publication amounts to 9%. Published theses com-

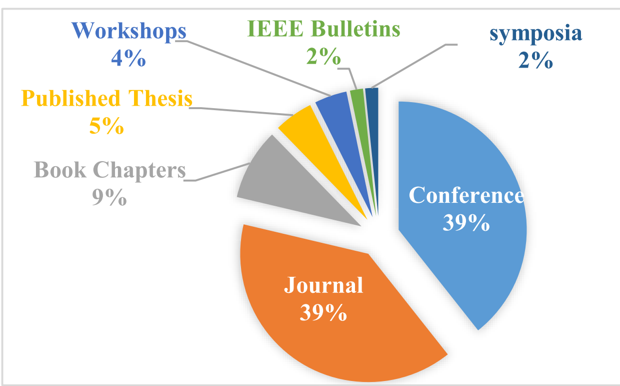


FIGURE 3. Percentages of selected studies' publication channels.

TABLE 4. RP stakeholders.

RP Stakeholders	Reference
Software architects and experts, product manager, customers	[80]
Analysts, customers	[81]
Customers and experts	[82]
Product managers	[83]
Users	[84]
Managers, software developers and customers	[85]
Software engineers, requirements engineers, customers and users	[22]
Group of customers and decision makers (project managers)	[10], [86]
Development teams	[87]
Marketing teams and developers	[88]
Developers	[76]
Product managers and customers	[21]
Technical and business experts	[61]
Software architectures and analysts	[67]
Experts (professional business analysts), requirements engineers and customers	[5]
Experts, (programmers) and customers	[89], [90]
Customers and design teams	[91]
Requirement specialists and groups of customers	[13][92]
Users, business analysts, customers and experts	[84], [87]
Project managers, requirements engineers and customers	[54], [93]
Business analysts and experts	[80], [94]
Architects	[38]
Customers and project managers and experts	[63]
Company executives, company vice presidents, project manager and marketing managers	[95]
Project managers, key customer representatives, development representatives	[52]
Analysts	[96]
Requirements engineers, marketing managers, professional team leaders, experienced end users	[8] [97]
System development teams, experts and customers	[4], [19]

prise 5%, followed by 4%, for the published workshops and 2% for symposia and IEEE bulletins.

A. RQ1 SIGNIFICANCE OF REQUIREMENTS PRIORITIZATION IN SOFTWARE DEVELOPMENT PROCESS

The main factor in determining the success of developed software systems is achieving and satisfying the expectations of stakeholders [10], [33]. Identifying the requirements that are most important to stakeholders from large numbers of elicited requirements is a challenge. Thus, identifying the most essential requirements is a major step towards software system success [6], [10], [26], [33], [34].

In industry, any development of a software system project usually encounters constraints in resources, including restricted human capital expertise, budget, technology and timelines [12], [16]. The RP process is an effective means of addressing this issue; it enables developers to deliver systems on time and ensures that stakeholders' needs are fulfilled within the time and budget constraints [10], [20], [21], [35]. The execution of the RP process produces an order list of the requirements that will be used by a development team for implementation in successive releases within resource constraints. Therefore, the chance of developing a successful system is increased because the most critical requirements are implemented and delivered first, thereby ensuring stakeholder satisfaction [20], [21].

Figure 4 shows the success percentage rates of system project development in 2015 from a recent study [36]. A total of 29% of all system projects were considered successful systems and delivered within project constraints, whereas 52% of projects were presented as challenging because they were not delivered on time, over the budget or/and did not implement sufficient features of the required stakeholders'

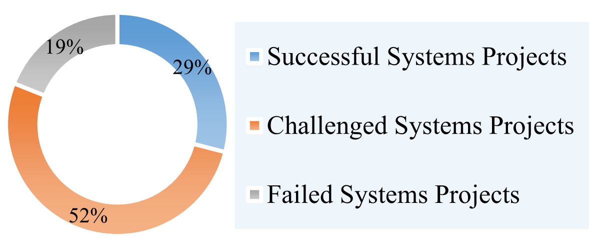


FIGURE 4. Success rates of the system project development in 2015.

needs. In addition, 19% of the projects were considered failed because of cancellation or delivery without use.

Complying with stakeholders' expectations, needs and time constraints are major challenges in producing successful systems [37]. The impact of these challenges can be reduced or eliminated with the execution of RP, which identifies the most important requirements to stakeholders and thus assists development teams in concentrating on delivering the most critical requirements [10], [38], [39].

Amongst the prime reasons behind the failure of software systems are lack of user involvement, lack of user expectations and short timelines [28], [29], [40]. RP can potentially help lower the risk of project cancellation and increase the success rate of projects when the stakeholders are actively involved [10], [20], [40], [41]. With RP, the order of requirement implementation is executed on the basis of the prioritized list of requirements, which is defined by the stakeholders' preferences.

Inevitably, RP also reduces the work effort because time is not wasted on implementing requirements that are not considered important to stakeholders. Thus, RP promotes plan stability [10], [42].Additionally, given that most organizations require their development teams to conduct cost–benefit analyses prior to undertaking any development project, RP can help optimize and manage resource usage [5], [10], [34].

Ignoring RP activity will lead to many challenges [10], [26]. For example, a project team may fail to satisfy customers mainly because deciding which requirements are essential to the customers will be difficult. This issue may lead to project failure because of the delivery of non-significant (i.e. 'nice-to-have' versus 'must-have') requirements to customers.

Furthermore, balancing the cost of each requirement against its business benefits without conducting the RP process will be challenging for the projects' stakeholders (especially the development team) [5], [16]. Often, the prioritization process will identify the priority value for each requirement according to its importance and cost involved. This process will help balance the benefit of each requirement and its cost. Consequently, the probability of producing a high-quality software system will be increased accordingly [5], [16].

B. RQ2 EXISTING RP TECHNIQUES AND THEIR PRIORITIZATION CRITERIA, TYPES, BENEFITS, SIZE OF REQUIREMENT SETS AND LIMITATIONS

RQ2 aims to identify and analyze existing RP techniques. Thus, the present techniques were identified. Each identified technique was critically analyzed on the basis of certain parameters: prioritization criteria (criteria that are used by each technique to prioritize requirements), limitations, types (execution type or automation level used by the technique to execute prioritization; three types are present: manual execution, which means manual performance of RP steps; semi-automated execution, in which one or more RP steps are automated and fully automated execution, wherein the entire RP process is fully automated), benefits and sets of requirements to prioritize. These sets are categorized into three: small (number of requirements < 15), medium ($20 <=$ number of requirements > 50) and large (number of requirements $>= 50$), as defined in [43].

From the selected primary studies of this review, 108 RP techniques were identified to be relevant to the current work. Figure 5 presents the outcome of classifying all the identified RP techniques. The RP techniques were classified according to execution type, which pertains to the automation level that is used to perform the prioritization process.

- Manual execution: These techniques perform all steps of the RP process manually.
- Semi-automated execution: These methods automate one or more steps in the RP process.
- Fully automated execution: These techniques provide full automation of the whole RP process.

From the defined classification types, 81 RP techniques were categorized as manual execution. These techniques require human experts to undertake the RP process manually, and no tool support is involved. A human expert refers to a person who has particular knowledge and relevant experience in related domains, such as software system development, project software practices and software marketing and business [5], [44], [45].

Twenty-five techniques were categorized under semi-automated execution. As the name suggests, some steps within the prioritization process are automated. The last category, fully automated execution, consists of two techniques that are implemented as automation tools (i.e. A Web-based Multi-criteria Decision-making Tool for Software RP [41] and ReproTizer [46]).

Table 6 in Appendix B depicts the result of analyzing the existing techniques in terms of prioritization criteria, benefits, limitations and size of requirement sets that are prioritized. Figure 6 presents the numbers of the techniques against the size sets of requirements. In the figure, 54 existing techniques are shown to prioritize a small set of requirements. The medium and large sets of requirements are considered by 14 and 13 existing techniques, respectively. Finally, 30 techniques have either not indicated the size of the requirement set or have not been adopted for prioritizing any set of requirements.

Our findings indicated that most of the existing techniques were applied to projects with small sets of requirements (i.e. not exceeding 20 requirements). In fact, these techniques do not sufficiently consider prioritising large numbers of requirements (partly because they are mainly evaluated as mere proof of concepts).

The existing RP techniques use various criteria for prioritizing the requirements, as shown in Table 6. The selection of the criteria in the prioritization process was based on the type and the aims of the technique for prioritizing the requirements. For instance, to produce an ordered list of requirements that is based on the required implementa-

FIGURE 5. **Classification of RP techniques.**

tion cost for each requirement, cost analysis is needed per requirement [16], [47]. Eighty-four (84) prioritization criteria were retrieved from the identified techniques.

Figure 7 depicts the reported prioritization criteria with their frequency usage from the identified techniques. The usage frequency indicates the number of

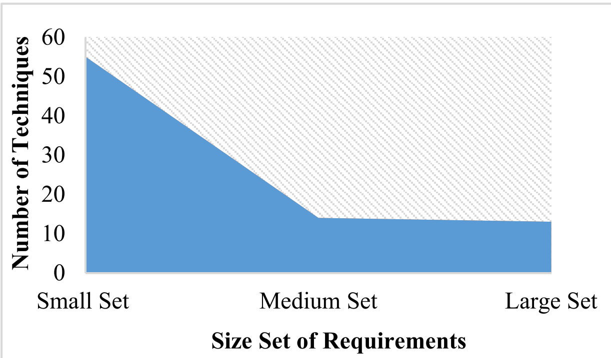


FIGURE 6. Techniques that address size of requirement sets.

times each prioritization criteria is used for prioritizing requirements.

Here, the importance of requirement implementation to stakeholders' criteria yielded a usage frequency of 51. The cost criteria amounted to a usage frequency of 22, whilst business value and value criteria had usage frequencies of 9. The dependency and risk had a usage frequency of 8 and 7, respectively. Next, the benefit and effort criteria indicated a usage frequency of 5 and 4, respectively. The penalty and software goal criteria had a usage frequency of 3. Furthermore, the business goal, completeness, modifiability, performance, schedule and time criteria produced a usage frequency of 2. Finally, the remaining criteria each showed a usage frequency of 1. Here, the important prioritization criteria were associated with the highest rank of the usage frequency in the existing techniques. This finding is in line with the need to prioritize requirements on the basis of their importance to stakeholders' needs.

As shown in Table 6, each technique has quality benefits and limitations. Scalability issues can be a limitation to many existing techniques. Scalability is the capability of these techniques to handle a large set of requirements. Most RP techniques cannot be implemented well with a large set of requirements. Only seven of 108 existing techniques can perform well with a large set of requirements: minimal spanning tree [21], HAHP [21], binary search tree [21], StakeRare [4], PHandler [5], requirements triage [48], clustering-based technique for large-scale prioritization [6], SNIPR [49], an optimal solution analysis technique for RP [50], ReproTizer [46] and RePizer [51]). However, these techniques suffer from other limitations, such as lack of automation of stakeholder quantification and prioritization (SQP), incapability to handle requirement interdependencies, overreliance on the participation of human experts and unreliable and poor fault tolerance, as depicted in Table 6.

SQP process plays a vital role as far as producing accurately and correctly prioritized lists of requirements is concerned [28], [29], [40], [52], [53]. Specifically, the SQP process identifies the degree of impact of each stakeholder on RP on the basis of its perceived importance [6], [38], [40], [54]. Table 6 shows that most existing techniques do not adopt a well-defined SQP for prioritization. Currently, requirement uncertainty prioritization approach (RUPA) [54], value-based intelligent RP (VIRP) [19], PHandler [5] and

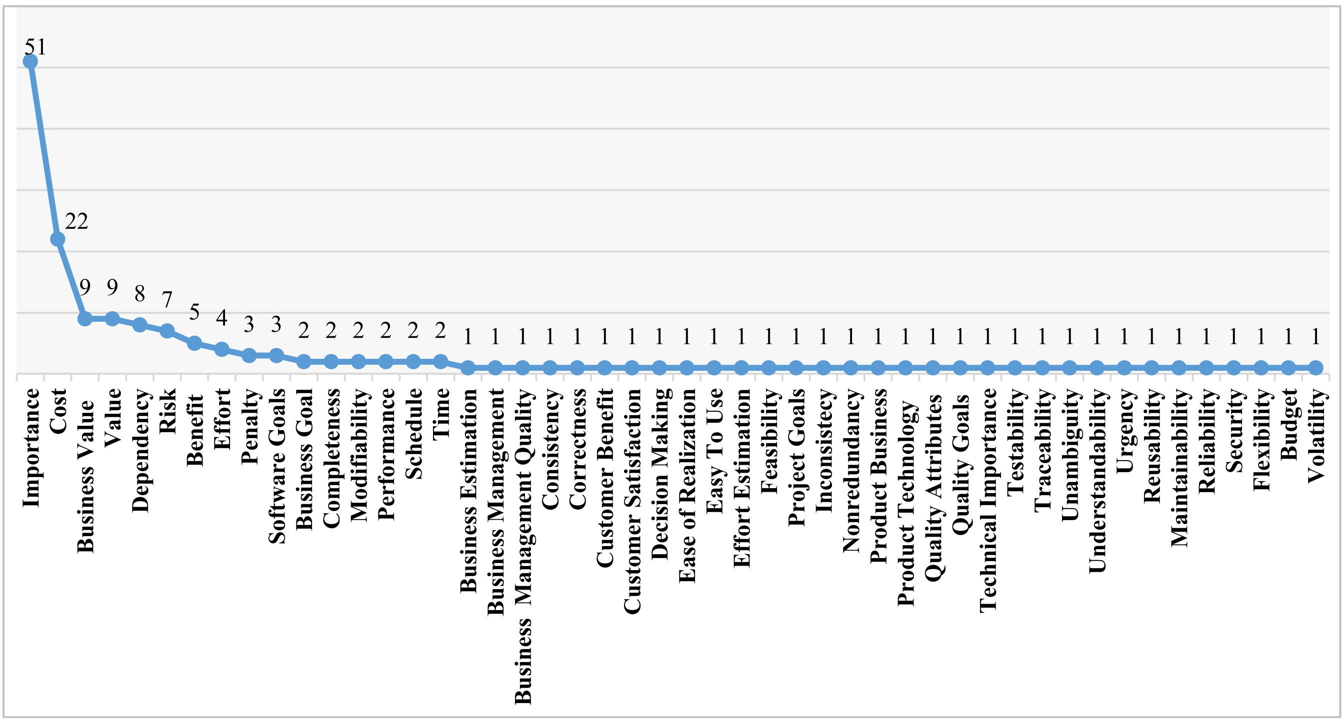


FIGURE 7. Requirements prioritisation (RP) criteria and their usage frequency.

StakeRare [4]? are amongst pioneer techniques that consider SQP in their prioritization process. In RUPA [54] and VIRP, the execution is based on manual adoption of the AHP-based method [19], [54]. In PHandler and StakeRare, the SQP is also executed manually; PHandler and StakeRare require more significant involvement of human experts to initiate the prioritization process than do RUPA [58] and VIRP [19].

Supporting fully automated prioritization may alleviate repetitive mundane processes, yet it must not hinder the judgment and creativity of human experts. However, excessive reliance on human intervention can also be counterproductive and lead to the following issues: human nature's bias parameters and unavailability of human experts [5], [55]. To minimize human intervention, AHP-based methods are adopted in undertaking the weighing process for stakeholders and requirements (e.g. Fuzzy AHP [56], [57], PHandler [5], RUPA [54] and hierarchical AHP [21]). However, employing the AHP-based method can introduce scalability issues (i.e. AHP-based methods do not work well with large projects that contain hundreds of stakeholders and requirements [26], [58], [59]).

Besides automation issues, the complexity of implementing prioritization can be problematic for some existing RP techniques, such as cost–value approaches [26], [60] and RUPA. For instance, the computational complexity of cost–value approaches increases with the increase in the number of requirements. In the case of RUPA, the employed IER algorithm is resource demanding because it adopts a computationally complex calculation [54]. Such complexity considerably affects time efficiency in the handling of large sets of requirements.

Handling requirement interdependencies is another important consideration in RP. Most existing RP techniques assume that all the requirements are independent and place the concerns of these interdependencies as future work. Not until recently, the need to cater to requirement dependencies during an RP process has emerged. This need has been addressed by multi-aspect-based RP [61], multi-decision-maker RP via multi-objective optimization [62], SNIPR [49], value-based RP [63], mathematical programming [64], RP under non-additive-value conditions [65], social network analysis for RP [66] and interactive RP [67].

In the case of multi-aspect-based RP, a dependency value for each requirement is identified independently by the stakeholders without any consideration of the potential dependencies of each individual requirement with others [61]. A similar approach is also adopted for multi-decision-maker RP via multi-objective optimization, SNIPR and value-based RP. In mathematical programming, the requirement interdependencies are randomly specified on the basis of the ratio of dependency amongst the requirements. The limitation of this approach is an ordering issue where the currently selected requirements cannot be implemented before the release of its predecessors. In RP under non-additive-value conditions [65], a directed cycle graph identifies and handles the requirement interdependencies. In social network analysis for RP, a matrix named as Req AND Req handles the dependencies during prioritization [66]. The dependencies of each requirement to others are identified by calculating the total degree centrality of each requirement (i.e. the degree to which the requirement is mostly required to other requirements), and then, the requirements are ranked in a descending order on the basis of centrality [66].

Finally, the interactive RP technique addresses the dependencies amongst the requirements by initially providing the list of the requirements with their associated prioritization classification level (i.e. low, medium, high) and the dependencies of the requirements that they have with one another. The priority level is induced by the dependencies amongst requirements [67].

C. RQ3 RP STAKEHOLDERS

The participation of the adequate stakeholders is crucial in producing an accurate RP result [6], [40], [54]. Thus, involving the right stakeholders is essential during prioritization [5], [29], [68]. The aim of RQ3 is to identify and categorize stakeholders involved in the RP process. Table 4 presents the stakeholders involved in the prioritization process according to the selected studies. Selecting stakeholders who participate in the prioritization is impacted by the criteria used to prioritize the requirements [10], [47]. For instance, if the requirements are prioritized on the basis of the importance and cost prioritization criteria, then the customers, project managers, requirements engineers and experts are chosen to participate [10], [47]. The customers then prioritize the requirements on the basis of the importance of requirements from a non-technical view. Simultaneously, the project managers, requirement engineers and/or experts evaluate the requirements on the basis of their technical knowledge (e.g. with respect to the cost of the prioritization criteria).

The findings in Table 4 indicate that the users and customers are highly participating stakeholders during prioritization. As such, most of the existing RP techniques aim to satisfy users and customers. This goal is realized by prioritizing the requirements on the basis of the importance criteria, which enables users and customers to specify the most essential requirements from their points of view. Meanwhile, the product manager, development teams (developers), requirement engineers/specialists, analysts and software architects are other stakeholders who participate in prioritizing the requirements on the basis of technical prioritization criteria (e.g. cost, time and penalty criteria). The experts and professional analysts are also involved in prioritization. They specify the priority value of each requirement. These priority values are assigned on the basis of technical prioritization criteria and the impact value for each participating stakeholder.

In accordance with our findings, the participating stakeholders' types could be categorized into three categories, as shown in Fig. 8: functional beneficiary, commercial and technical stakeholders. The functional beneficiary stakeholders include stakeholders who foresee other stakeholders' satisfaction (e.g. the importance of requirements for system

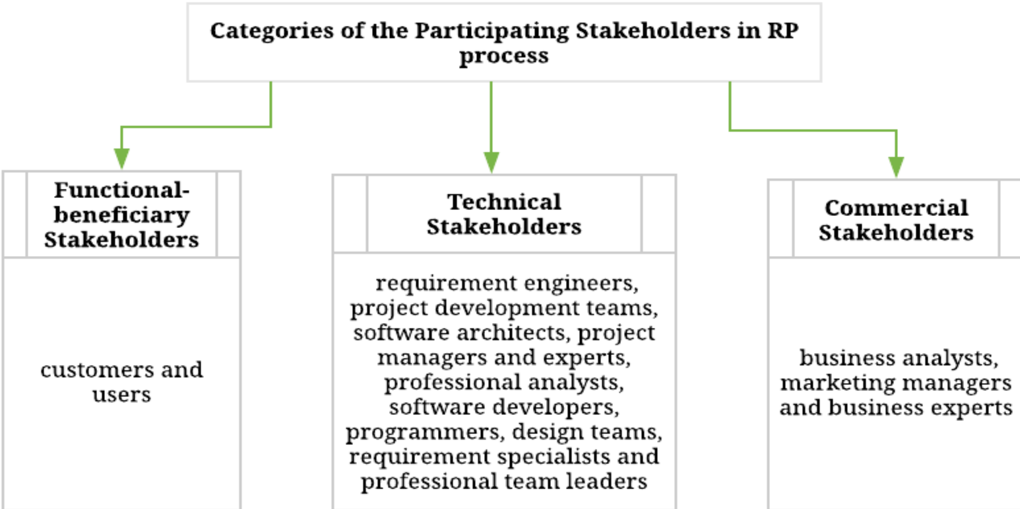


FIGURE 8. Categories of RP stakeholders.

functionalities). Typically, these stakeholders are those who pay for the system, specify the system functions to be developed and use its services. They are the customers and users of the system.

The commercial stakeholders are those who participate in terms of commercial prioritization criteria, such as product business, commercial goals, business estimation and business management. These stakeholders are business analysts, marketing managers and business experts. The technical stakeholders comprise stakeholders who participate on the basis of the technical prioritization criteria, such as dependency, efforts, time and cost criteria. These stakeholders include requirement engineers, project development teams, software architects, project managers and experts, professional analysts, software developers, programmers, design teams, requirement specialists and professional team leaders.

D. RQ4 RP USAGE CONTEXTS

Answering RQ4 involved two stages. Firstly, the contexts of the identified techniques and selected studies were revealed. Secondly, the usage frequency percentage of each context in the existing techniques was quantified. The usage frequency was measured by the number of times the RP techniques were proposed or applied in each identified context. As such, we categorized the selected studies and the study focus in accordance with the publication years and their contexts, respectively. The implication of the RP process through the identified techniques concerns few contexts of software development. These contexts are software release planning (SRP), agile software development (ASD), value-based software development (VBSD), software architecture (SA), social network system development (SNSD), real-client custom development projects (RCCD), cognitive driven (CD), system development organizational work (SDOW), market-driven software development (MDSD) and goal-oriented RE (GRQE). We also included an additional context (not specified [NS]) to include cases where the RP technique does not specify any context.

Figure 9 presents the usage frequency percentage of each context of the listed RP techniques. Most of the RP studies (49%) provides their RP processes for general software development (as NS context). ASD context had a usage frequency of 21%, followed by SRP with 16%, VBSD with 4%, SA, MDSD and GRQE with 2% and finally CD, SNSD, RCCD and SDOW with 1%. Placing the NS context aside, SRP and ASD yielded the top usage contexts regarding RP. This finding relates to the benefits of the environment development process within the ASD and SRP contexts. In the SRP context, the identification of the core requirements is often strictly observed within the projec's constraints. As such, conduct of the RP process is necessary to select the most important requirement to be delivered [69], [70]. Similarly, prioritizing the requirements is a crucial step within the ASD context. In this context, the RP process is conducted to ensure that correct requirements are selected and included in each iteration during prioritization [18], [71].

In addition to Fig. 9, Fig. 10 illustrates the publication tendency of the selected primary studies on the basis of their study contexts across the publication years between 1993 and 2018. In general, the number of the published studies related to RP was increasing yearly. This increase could be observed from 2005 to 2018, with a noticeable peak in 2012. This result indicated strong interests on applying RP in their relevant contexts (such as ASD, SA, GRQE and VBSD) by RP researchers and practitioners. Publication within the NS context started in 1994, with a peak in 2010. This result indicated a healthy improvement, especially in the general awareness of the RP process.

Meanwhile, the focus of the selected studies on the ASD context started from 2008, coinciding with the practices of the

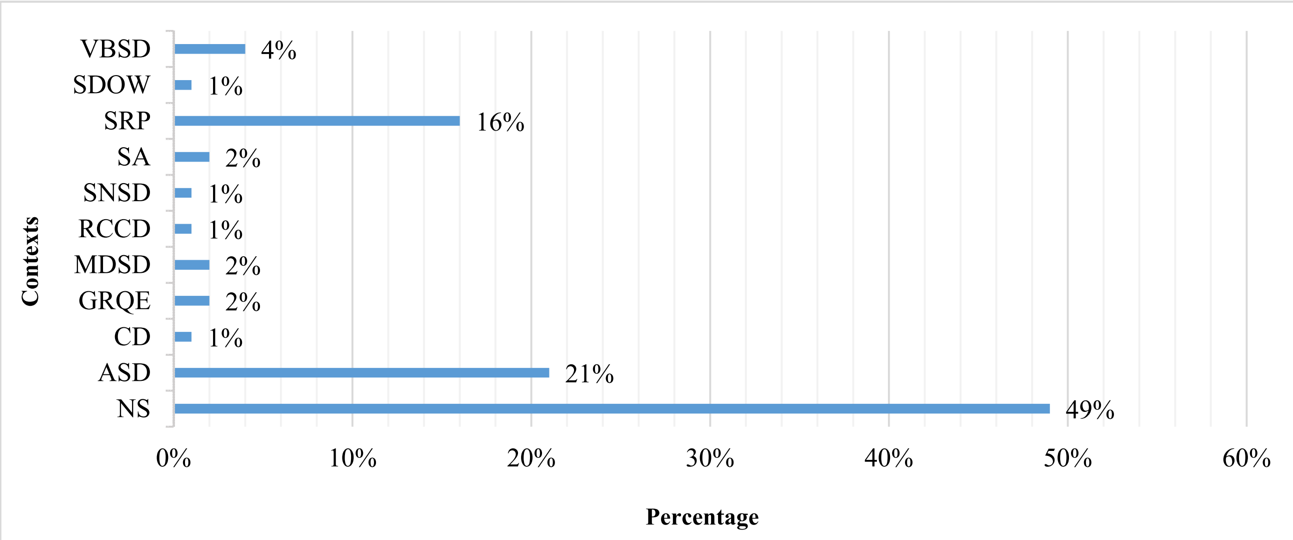


FIGURE 9. Contexts of RP techniques.

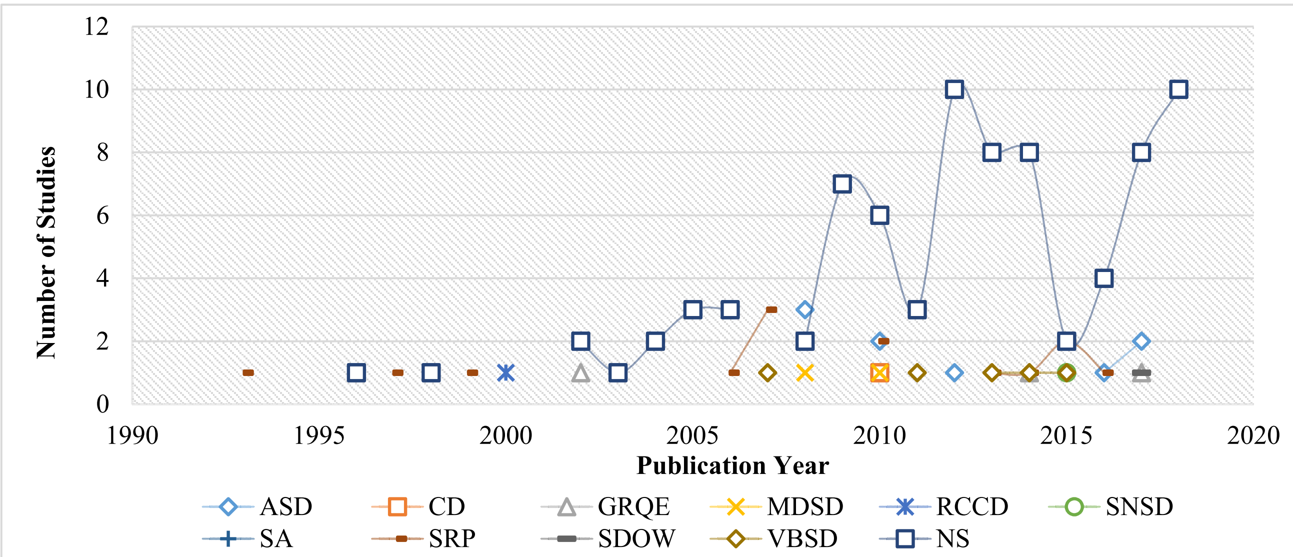


FIGURE 10. Publication tendencies of selected primary studies by publication year and study focus with contexts.

agile development process [72]–[74], which began in 2005. The publication of the selected studies on ASD context was steady until now due to the popularity of the agile development process. Regarding the SRP and VBSD contexts, the ranges of publication were from 1993 to 2016 and from 2007 to 2015, respectively. Regarding the GRQE context, the publications were sparsely distributed in 2002, 2014 and 2017. This result indicated little work in the field of GRQE. Similarly, the published studies within the context of MDSD were also sparsely distributed in 2003 and 2010. Finally, the SA, CA, RCCD, SDOW and SNSD contexts each had one published study in 2013, 2010, 2000, 2017 and 2015, respectively.

E. RQ5 RECOMMENDED FUTURE SETS OF RP

The RP domain, particularly in terms of the development of context-based RP techniques, has limitations, strengths and opportunities. Although the right sequence of requirement selection is ensured, RP implementation can be expensive and resource hungry (especially when the number of requirements increases). Issues of scalability, lack of SQP process, the need for involvement of professional analysts in RP and SQP, requirement dependencies, complexity, lack of automation level and time factors are common challenges in the RP processes of existing RP techniques, as summarized in Table 6 and further elaborated in the previous sections. Therefore, novel opportunities are provided to enhance

TABLE 5. Quality scores of selected studies.

TABLE 5. *(Continued.)* Quality scores of selected studies.

Reference	QAC1	QAC2	QAC3	QAC4	QAC5	QAC6	Overall Score/12
[4]	2	2	2	2	2	2	12
[5]	2	2	2	2	1	2	11
[6]	2	2	2	2	1	2	11
[8]	2	2	2	2	1	2	11
[10]	2	2	2	1	0	1	8
[11]	1	1	2	1	0	1	6
[12]	1	2	1	1	1	1	7
[13]	2	1	2	1	1	1	8
[14]	2	2	2	2	2	1	11
[15]	2	2	2	1	0	2	9
[16]	2	2	2	1	0	2	9
[19]	2	1	2	2	2	1	10
[20]	2	2	2	2	1	2	11
[21]	2	2	2	2	1	2	11
[22]	2	2	1	2	1	2	10
[23]	2	1	2	1	1	2	9
[26]	2	2	2	2	0	2	10
[29]	2	2	1	2	2	1	10
[33]	2	2	2	2	1	1	10
[34]	2	1	2	1	1	1	8
[35]	2	1	1	1	0	1	6
[37]	2	2	2	1	0	1	8
[38]	2	2	2	1	1	1	9
[40]	2	2	2	2	2	1	11
[41]	2	2	2	1	1	2	10
[42]	2	1	2	2	1	2	10
[43]	2	2	1	1	1	2	9
[46]	2	2	2	2	2	1	11
[47]	2	2	1	1	0	1	7
[48]	2	2	2	2	1	1	10
[49]	2	2	2	2	1	1	10
[50]	2	2	2	2	2	1	11
[51]	2	2	2	1	1	1	9
[52]	2	2	1	1	2	2	10
[54]	2	2	2	2	1	1	10
[55]	2	1	1	1	0	1	6
[59]	2	1	2	1	0	1	7
[60]	2	2	2	2	1	1	10
[61]	2	2	2	2	1	1	10
[62]	2	1	2	2	0	0	7
[63]	2	1	1	1	0	1	6
[64]	2	2	2	2	1	1	10
[65]	2	2	1	1	0	0	6
[66]	1	1	2	2	0	0	6
[67]	2	2	2	2	2	1	11
[71]	2	1	1	2	0	0	6
[75]	2	1	1	1	1	1	7
[76]	2	2	2	1	1	1	9
[77]	2	2	1	0	0	1	6
[78]	2	2	1	1	1	1	8
[79]	2	1	1	1	0	1	6
[80]	2	2	1	2	1	2	10
[81]	1	2	1	1	1	1	7
[82]	2	1	2	1	0	1	7
[83]	2	2	1	2	1	1	9
[84]	1	2	2	1	0	1	7
[85]	2	2	1	2	1	2	10
[86]	1	1	1	1	1	1	6
[87]	1	1	2	1	0	1	6
[88]	2	1	1	1	0	1	6
[89]	2	2	2	1	0	2	9
[90]	2	2	2	1	1	2	10
[91]	2	2	2	1	1	1	9
[92]	2	2	1	1	1	2	9
[93]	2	1	2	2	1	1	9
[94]	2	1	1	1	1	0	6
[95]	2	1	2	1	0	0	6
[96]	1	1	2	1	1	1	7
[97]	1	1	2	1	1	1	7
[98]	2	2	2	2	1	1	10
[99]	2	1	2	2	1	1	9
[100]	2	1	1	2	2	1	9
[101]	2	2	1	1	1	2	9
[102]	2	2	2	1	2	2	11
[103]	2	2	2	2	1	1	10
[104]	2	1	2	1	1	1	8
[105]	2	1	2	1	1	1	8
[106]	2	2	1	2	2	2	11
[107]	2	1	1	1	1	1	7
[108]	2	1	1	1	1	1	7
[109]	2	2	1	2	1	2	10
[110]	2	2	2	2	1	1	10
[111]	1	1	1	1	1	1	6
[112]	2	1	2	1	1	1	8
[113]	1	2	2	1	1	1	8
[114]	2	1	2	1	0	1	7
[115]	2	1	1	1	0	1	6
[116]	2	2	2	2	1	1	10
[117]	2	1	2	1	0	1	7
[118]	2	1	1	1	1	1	7
[119]	2	1	2	1	1	1	8
[120]	2	1	2	2	1	1	9
[121]	2	1	2	1	1	1	8
[122]	2	1	2	2	2	1	10
[123]	1	1	2	1	1	1	7
[124]	1	1	2	2	0	0	6
[125]	2	1	1	2	1	1	8
[126]	2	1	2	1	1	1	8
[127]	2	1	2	2	1	1	9
[128]	2	1	1	2	0	0	6
[129]	2	1	1	2	1	1	8
[130]	1	1	2	2	1	0	7
[131]	2	2	2	2	1	1	10
[132]	2	1	2	2	0	0	7
[133]	2	1	1	1	1	0	6
[134]	2	2	2	2	1	1	10
[135]	2	2	1	2	1	1	9
[136]	2	2	2	2	2	1	11
[137]	2	1	2	2	1	1	9
[138]	2	2	1	2	1	1	9
[139]	2	2	2	2	1	1	10
[140]	2	1	1	2	1	1	8
[141]	2	1	1	1	1	1	7
[142]	2	2	1	2	1	1	9
[143]	2	1	2	2	1	1	9
[144]	2	2	2	2	1	1	10
[145]	1	1	2	1	1	1	7
[146]	2	1	2	1	1	1	8
[147]	2	2	1	2	1	2	10
[148]	2	1	2	1	0	0	6
[149]	2	1	2	2	1	1	9
[150]	2	1	2	1	1	1	8

existing RP techniques by addressing these identified issues. These opportunities can be achieved by introducing new methods or enhancing these RP techniques. These methods can add value to the RP domain with the following points.

- Automation of RP process phases by assigning priority values to requirements and minimizing pairwise comparisons amongst requirements. This issue can be addressed by using machine learning techniques, such as case-based reasoning, reinforcement learning and deep learning.
- Maximization of the involvement of professional analysts and of the role of experts in creative activities, and the minimization of their involvement in mundane (and automatable) aspects (e.g. quantification of requirements and setting priority values), which are subjected to human bias with little availability of experts.
- Adoption of scalable and robust (big data related) algorithms that can work well for projects with large sets of requirements. Potentially, such adoption enhances efficiency of the time performance and scalability of the proposed technique.

V. THREATS TO VALIDITY

Systematic review researches are often subjected to four different types of threats to validity (i.e., conclusion, internal, construct and external validity).

As the name suggests, threats to conclusion validity involves the potential issues that affect the conclusion as the result of the improper treatment of the variables of interest against the outcome. One potential threat to the conclusion validity relates to the bias in the selection of relevant studies and data synthesis. To mitigate this threat, a precise study selection strategy was designed on the basis of [27], which includes the inclusion and exclusion criteria and extensive QAC. This strategy was applied precisely to validate the appropriateness of each included study. Also, the data were critically extracted and then qualitatively and quantitatively synthesized to obtain data from the selected primary studies. Additionally, a set of particular QAC was applied to prevent imprecise inclusion. However, we still cannot guarantee that the defined inclusion and exclusion criteria, QAC and data synthesis are sufficient to repress the threats of bias in selecting the relevant studies and data synthesis of this review.

Threats to internal validity relates to the issues that concerns with the relationship of the variables of interest and the outcome. In the current study, the performance of a particular RP can be dependent on the scale of project undertaken. Some RP techniques are applied only to small projects by choice (i.e., sometimes owing to the limited available case studies at the time). As a result, their true performance for large projects might not be fairly evaluated. To mitigate this issue, we have considered multiple sources of publications of the same work (e.g from relevant journals, book chapters and conferences) whenever possibl

Construct validity threats concern on the relationship between the application and theory. One threat to the construct validity comes from the exclusion of the potential relevant studies. To repress this threat, a rigorous search strategy was defined and used reduce the threat to completeness of retrieval and to include all the studies relevant to the RP domain. Consequently, website articles, studies in progress, research published in non-peer-reviewed publications, collectively called grey studies, were excluded. These excluded studies may provide answers to any of the research questions. As such, potential relevant studies might have been missed.

Finally, threats to external validity involve the issues that limit the ability to generalize the SLR findings outside the study scope. As mentioned in the defined study selection strategy, the non-English studies and grey studies were excluded. The threats here establish in whether the final selected studies of this review are able to constitute all types of review studies in field of the RP. We believe that our constructed review protocol assisted us to select a typical set of studies that can include the domain knowledge of the former researches and provide a comprehensive source of data and information for practitioners and researchers who are working in the field of RP. However, our results in this SLR are more concern on the RP domain from the academic perspectives than the industrial environments. Additionally, to be specific, there might be grey issues related to next release problem that have not been considered (e.g. versioning of artifacts and configuration managements) as they do not impact requirements prioritization directly.

VI. CONCLUSION

This research provides a comprehensive review of the RP domain by identifying its significance in the system development process and the stakeholders involved in the RP process and then categorizing them on the basis of the RP software development contexts, the existing techniques of RP, the limitations and the future sets.

In this SLR, a review protocol was constructed on the basis of the standard SLR guidelines of Kitchenham and Charters [27]. The researchers formulated search strings that are based on the stated research questions. In the initial phase, 878 prospective research studies were derived. With the execution of the defined selection strategy, 122 relevant studies were finally selected.

The results of the study presented the significance of RP and disclosed the types of stakeholders who participate during the prioritization process, the prioritization criteria, the suitable set of requirements, the benefits and the challenges of each RP technique in detail.

In summary, the findings revealed that limitations still exist despite the presence of existing RP techniques. These limitations include complexity, scalability, lack of automation and intelligent terms, dependency, lack of quantification and prioritization for the participating stakeholders and need for substantial professional involvement. If RP techniques are to be pervasively adopted in the field of RE, there is a strong need to adopt the identified current strengths and address their inherent limitations. Thus, the current work presents a small leap forward that can serve as guidelines for future development of RP techniques.

TABLE 6. Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
1.	A Conceptual Model And Process For Client-Driven Agile RP [87]	Business value, risk and effort estimation	NS			*	It requires the involvement of experts to conduct the prioritisation process.	NS
2.	Adaptive Fuzzy Decision Matrix Model for RP [121]	Reusability, flexibility, performance and maintainability	Small	*		*	It is heavily reliant on expert participation to initiate and conduct prioritisation.	It can consider more than one prioritisation.
3.	Adaptive Fuzzy Hierarchical Cumulative Voting [123]	NS	Small		*	*	NS	NS
4.	AHP [10], [15], [21], [22], [92], [102], [116]	Importance, cost and penalty	Small	*	*		It is complex and has a large number of requirements. Its speed is slow.	It produces accurate results and is suitable for use with small sets of requirements.
5.	AHP_GORE_PSR [119]	NS	Small	*			It needs to be tested with a real project dataset to allow for performance evaluation.	NS
6.	Analytical Model for Requirements Selection Quality Evaluation [125]	Importance	Medium				The analytical model used is limited by approximations required by the technique, which affects the prioritisation performance efficiency in terms of accuracy.	NS
7.	ANN Fuzzy AHP Model [146]	NS	NS			*	It heavily relies on the involvement of experts to initiate the prioritisation process It does not consider the dependencies among the requirements	NS
8.	Approach for RP based on Tensor Decomposition [142]	Importance, business values, cost and risk	Small	*		*	It is not catering the dependencies among the requirements It has not been implemented and evaluated with large set of requirements	It is less time consumption than AHP method
9.	A priori technique [127]	Value	Small				The technique has been only applied to a small set of requirements manually.	It introduces a parallelised and implementable method in addressing the stakeholder conflicts in the RP process.
10.	A Preference Weights Model For Prioritising Software Requirements [20]	Importance	Small			*	The proposed technique needs to be tested with a large set of requirements. It needs to reduce the disagreement rate amongst final rank weights.	It is a fast and fully automated technique.
11.	A Web-based Multi-criteria Decision-making Tool For Software RP [41]	Importance	Small	*		*	Rank reversals are present. It needs to test the developed tools with a large set of stakeholders and requirements to validate its effectiveness and performance.	NS
12.	Architecture Trade-off Analysis Method [80], [151]	Quality goals and dependency	NS		*	*	It requires detailed technical knowledge to execute the technique. It does not address requirement interdependencies.	Scenario generation based on requirements is suitable for dynamic and static properties.

TABLE 6. ***(Continued.)*** **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
13.	Attributed Goal-oriented Requirements Analysis Method [81]	Correctness, unambiguity, completeness, inconsistency, modifiability and traceability	NS			*	It is heavily reliant on the involvement of expert analysts in execution.	NS
14.	Benefit and Cost Prediction [82]	Importance	NS			*	It does not provide priority value for each requirement and does not handle the dependencies amongst the requirements.	It can handle the costs and benefits of the requirements.
15.	Binary Priority List [26], [83]	Benefit	Small	*		*	It considers only one type of stakeholder (product manager) and only one aspect (benefit).	It can be used as a sorting algorithm for requirements.
16.	Binary Search Tree [21], [37], [89], [105], [112]	NS	Large			*	It provides a simple ranking without assigning priority values to requirements.	Its implementation is simple. It is usable for large numbers of requirements.
17.	Binary Tree [55], [105]	Importance	Small	*		*	It is complex and does not scale well. It is not implemented and tested in a collaborative environment.	It can be used as a sorting algorithm for other RP techniques.
18.	Bubble Sort [5], [21], [26]	NS	Small	*	*	*	It provides a simple ranking and does not assign priority values to requirements.	It can be used as a sorting algorithm for other techniques.
19.	Case Based Ranking [26], [98]	Importance	Small, Medium	*		*	It cannot support coordination in negotiations amongst stakeholders.	It uses machine learning to reduce the amount of information required from the stakeholders.
20.	Clustering-based Technique For Large Scale Prioritisation [6]	Importance	Large				It requires the expert to implement the prioritisation process. It needs to be tested with additional projects for its results to be generalised.	It can handle a large set of requirements.
21.	Cognitive Driven RP [109]	Importance	Small	*		*	It does not address the dependencies amongst the requirements.	It improves stakeholders' negotiation by reducing misunderstandings.
22.	Conceptual Model of Agile RP [84]	Importance and business value	NS				NS	NS
23.	Correlation-based Priority Assessment Framework [85]	Business estimation, management and quality	Small				It does not consider the negative correlations amongst requirements in the priority assessment. It needs to be tested with a large set of requirements to evaluate its results.	It is helpful in understanding the relationships amongst requirements from multiple perspectives.
24.	Cost–benefit Analysis Method [80]	Cost, benefit and schedule	NS				Well-defined steps of quantifying benefits and costs are absent.	It provides business measures for particular system changes.
25.	Cost- value Approach [22], [26]	Importance and cost	Small	*	*	*	Involvement of professional business analysts (experienced software engineers) is required to execute the prioritisation process. Complexity increases with large/medium sets of requirements. It ignores requirement interdependencies.	It is a clear and usable method. It can integrate judgments of cost and value of requirements that are considered for implementation.

TABLE 6. *(Continued.)* **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
26.	Cumulative Voting (CV) [10], [75], [80], [86], [114], [117]	Importance	Small	*		*	It does not permit the evaluation of the relative priority difference amongst the requirements.	It is fast and simple.
27.	Dot Voting [84], [87]	Importance	NS			*	It ignores requirement interdependencies.	NS
28.	DRank [8]	Dependency and importance (stakeholders preferences)	NS				It requires professional intervention to perform the prioritisation process.	It considers business dependencies amongst the requirements. It provides an easy-to-use method of selecting the prioritisation criteria.
29.	Eclipse Process Framework [87]	Easy to use	NS				NS	NS
30.	Evolve [11], [26], [60]	Risk, benefit resources, business value and effort	Small	*	*		It is computationally complex.	The evaluation and the identification of benefits are associated with different release plans.
31.	Exploiting User Feedback in Tool-supported Multi-criteria RP [128]	Value	NS			*	It needs to be validated in real scenarios for performance assessment. The decision makers are a prominent source of information for prioritising the requirements, which can affect the prioritisation process with biases induced by the decision makers.	It can consider the feedback of different stakeholders, such as the users and decision makers, in the prioritisation process.
32.	Fuzzy AHP [26], [110], [113]	Software goals	Small	*		*	It needs to be tested with large projects and does not handle the dependencies amongst requirements.	NS
33.	Fuzzy-based MoSCoW [129]	Importance	Small				It cannot prioritise the requirements with participation of more than one stakeholder and handle the requirement dependencies in the prioritisation process. The technique needs to be validated with large numbers of requirements for the performance to be generalised.	NS
34.	Fuzzy Hierarchical CV [118] [137]	Importance	Medium			*	It has a high-risk decision.	NS
35.	Fuzzy Technique to RP [110]	Importance and software goals	Small			*	It has only been applied to a small set of requirements It needs to be applied in real scenarios.	NS
36.	Fuzzy Multi-criteria Decision-making Approach for software RP [38]	Importance	Small	*		*	It needs to be implemented and tested with additional projects, especially those with large sets of requirements, for performance validation.	NS
37.	Goal Oriented Approach for Software Requirements Elicitation and Prioritisation [126]	Cost and effort	Small	*			NS	NS
38.	Graph-oriented Requirement Selection [131]	Cost and value	Small	*		*	NS	It considers the value-related dependencies in the selection process of the requirements
39.	Handling uncertainty in agile RP and scheduling [34]	Business Value	Small	*			It needs to be validated with large scale of requirements and lack of handling the dependencies amongst the requirements.	NS
40.	Hierarchical CV (HCV) [103], [106]	Importance	Medium			*	It does not cater the requirements interdependencies.	Scalability and resulting priorities are improved with HCV compared with those of CV.

TABLE 6. ***(Continued.)*** **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
41.	Hierarchical AHP [21], [26], [58]	Importance	Medium /large		*		It is hard to be applied. It produces many judgment errors because it cannot address consistency, as in the case of AHP. It is unreliable and has poor fault tolerance.	It reduces the number of comparisons in the AHP technique.
42.	Hybridized technique for RP [122]	Importance	Small				It has been applied to small-scale requirements only.	NS
43.	Integrated Prioritisation Approach [136]	Importance	Small	*		*	It needs to be adopted in a controlled experiment in a real scenario setting to explore the technique performance and compare the experiment findings with those of other studies.	It can prioritise functional and non-functional requirements and consumes less time than does AHP.
44.	Interactive RP [67]	Importance and dependency	Medium				It needs to be tested in a complex industrial situation.	NS
45.	Kano Model [84], [87]	Importance	NS			*	NS	NS
46.	Lanchester Theory [88]	Customer satisfaction with technical excellence	Small	*			It does not set relative values for the linguistic terms, which can assist the relative weight calculation across all the requirements. It needs to be implemented with a large set of requirements for performance validation.	NS
47.	Market Driven RP model [97]	Importance	Small	*		*	It does not address the requirement interdependencies in the RP process, and its efficiency must be evaluated with real and large industrial projects.	NS
48.	Mathematical Programming Technique [76], [64]	Dependency in terms of influence and cost	NS				NS	It considers requirement interdependencies and SQP.
49.	MoSCoW [43], [111], [117]	Importance	NS			*	It does not assign priority values to requirements and ignores the requirement interdependencies.	NS
50.	Minimal Spanning Tree [21], [23], [26]	NS	Medium /large		*	*	It is sensitive to judgment error as all redundancy is eliminated. It is unreliable and has poor fault tolerance.	It eliminates the redundancy of the pairwise comparison of AHP. It is suitable for large sets of requirements if reliability and fault tolerance are not important.
51.	Multi-aspects Based RP Technique [61]	Business and technical aspects	Small	*		*	It is heavily dependent on the involvement of experts to execute the RP process. Its performance must be evaluated with a large set of requirements.	NS
52.	Multi-criteria Preference Analysis Requirements Negotiation (MPARN) for RP [100], [102]	NS	Small			*	It does not detect inconsistencies amongst ranking values.	NS
53.	Multi-objective Next Release Problem for RP [67], [140]	Value and cost	Large				It does not produce an ordered list of requirements as final result and instead groups requirements for the planning of subsequent releases. It also does not handle requirement interdependencies.	NS
54.	Multi-objective ant colony optimisation for requirements selection [135]	Importance, cost and dependency	Large				The performance efficiency of the techniques must be evaluated to reveal its efficiency in terms of accuracy, complexity and time consumption.	It can address requirement dependencies.

TABLE 6. ***(Continued.)*** **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
55.	Multi-decision-maker RP Via Multi-objective Optimisation [62]	NS	NS			*	The prioritisation process heavily relies on the participation of decision makers, and technique performance has not been evaluated.	It handles requirement dependencies.
56.	Multi-voting System for RP [84], [87]	Importance	Small	*		*	It cannot be executed for more than 20 requirements.	NS
57.	New approach for RP [143]	Value and cost	Small	*		*	It has been tested with small set of requirements and lack of evaluating the importance of the participating stakeholders	NS
58.	New Strategy for Prioritising Functional Requirements [145]	Dependency	Small	*		*	It does not evaluate the impact of the participating stakeholder in the prioritising the requirements	NS
59.	Numerical Assignment (NA) [5], [10], [26], [92], [104]	Importance and time	Small				It cannot provide a definition of a ranking list. It does not provide priority values for requirements and standard descriptions of specified categories to stakeholders.	It is easy to use. It is relatively straightforward.
60.	Optimal Solutions Analysis technique for RP [50]	Cost and value	Large				It cannot handle requirement interdependencies and heavily relies on the involvement of experts, which may induce human bias.	It considers SQP and can work with large sets of requirements.
61.	Pair-wise Comparison [80], [90], [92], [101]	Importance	Small	*	*		It is complicated and produces unreliable results.	It can work with various comparison criteria that are based on the decision makers.
62.	Partial Order Assimilation Approach for RP [118]	Decision making	Medium				It cannot combine stakeholders' feedback when two or more important ranking functions are present. It cannot handle requirement dependencies.	NS
63.	Performance Enhancement in RP by Using Least-squares-based Random Genetic Algorithm [150]	Cost	Small			*	It has been applied to the small set of requirements. It lacks of addressing the dependencies among the requirements during the prioritisation process	It can performs the prioritisation process with less time consumption than AHP technique
64.	PHandler [5]	Importance	Large				It requires the involvement of professional business analysts and experts to perform the RP process and needs AHP to be performed and merged with requirement values. It does not consider requirement interdependencies.	It can work with a large set of requirements. It considers SQP in its prioritisation.
65.	Ping Pong Balls [84], [87]	Importance	Small				NS	NS
66.	Planguage [98]	Stakeholders' goal	Small	*	*	*	It uses basic ranking and does not provide relative differences amongst ranks.	NS
67.	Planning Game[5], [26], [59], [75], [89], [90], [101]	Importance	Small	*		*	It does not identify the priority value of each requirement.	It is suitable for innovative development models, such as extreme program, because of its flexible behaviour.
68.	Planning Game Combined With AHP [80], [90], [114]	Importance	Small	*		*	The RP process is highly reliant on experts' involvement.	NS
69.	PRFGORE [132]	Performance, security and reliability	Small			*	It lacks tools to support the participating decision makers in conducting the prioritisation process. It does not address requirement interdependencies. The effectiveness of the proposed technique has not been empirically evaluated with real industrial projects.	NS

TABLE 6. ***(Continued.)*** **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
70.	Prioritisation Analysis For RP [147]	Importance	Medium			*	It does not address the requirements interdependencies	It provides iterative procedure that can be used to identify, select and prioritise requirements of the user.
71.	Priority Groups [16], [38]	Importance	NS	*	*		It is not easy to use, is unreliable and has poor fault tolerance.	NS
72.	Prioritising Requirements in Agile Development [71]	Cost, risk and schedule	NS				It has not been implemented and evaluated in empirical experiments with real projects.	It can handle the factors concerning the effectivity of the RP process.
73.	Quality Functional Deployment QFD [14], [91], [107]	Technical importance	Small	*		*	NS	NS
74.	Quantitative WinWin [138], [139]	Importance, effort, business value and time	Small				The prioritisation is based on the availability of a precise model for the effort estimation and willingness of the participating stakeholders in eliciting their preferences. Its scalability needs to be assessed with large sets of requirements.	It can provide quantitative analysis for good decisions in selecting the core requirements.
75.	Ranking [10], [13], [92]	Importance	Small	*		*	It cannot align many stakeholders' preferences and does not handle requirement dependencies.	It is convenient technique with participating only single stakeholder
76.	Ranking Based on Product Definition [84], [87]	Product criteria: business and technology	NS				The execution of the RP process is heavily reliant on the involvement of professional analysts.	NS
77.	RIZE [130]	Business value, cost, risk and volatility	Small				An empirical evaluation is needed to assess the efficiency of the technique.	It is easy to understand and use.
78.	Relative Weighting [87]	Importance	NS			*	It is heavily reliant on the involvement of experts to perform RP.	NS
79.	ReDCCahp [144]	Cost and value	Medium	*		*	It does not cater the requirements interdependencies in the prioritisation process	It is more effective than AHP in prioritising medium set of requirements with less complexity
80.	RePizer [51]	Cost	Large	*		*	It requires participating experts to execute the prioritisation and does not consider requirement dependencies.	NS
81.	ReproTizer[46]	Importance	Medium /Large			*	It does not address requirement interdependencies.	Its time utilisation is effective.
82.	Requirement Triage [48]	Business goal	Large			*	It does not recall results and is prone to error.	NS
83.	RUPA [54]	Importance	Small	*	*		It lacks automation and intelligent terms. It relies on the involvement of professional business analysts to perform RP, and it has computational complexity.	It provides a priority value for each requirement. It considers SQP.
84.	Round the Group Prioritisation [84], [87], [108]	Importance	Small			*	It is heavily reliant on the participation of professional analysts in RP.	NS
85.	RP Solution Model [148]	Cost, dependency and benefit	NS			*	It provides only the conceptual procedure for performing the RP process without implication it with the real scenarios	NS

TABLE 6. *(Continued.)* **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
86.	RP under Non-additive Value Conditions [65]	Dependency, value and cost	NS			*	It has not been validated with real industrial projects.	It considers the dependencies amongst the requirements during prioritisation.
87.	RP_WOA [141]	Cost and value	Medium			*	There is a need to validate the accuracy performance of the technique It lacks of considering the requirements independencies	Ability to prioritise the requirements with efficient time consumption performance
88.	SAFFRON : A Semi-automated Framework for Software RP [134]	Importance	Small	*			The RP execution heavily depends on experts' participation and does not address requirement interactions.	It updates the prioritisation lists whenever new requirements are elicited.
89.	Simple Multi-criteria Rating Technique By Swing for RP [78], [102]	NS	NS	*	*	*	It does not detect inconsistencies amongst ranking values.	NS
90.	SNS Technique [133]	NS	Small			*	It is not specifically concerned about presenting the set of prioritised requirements and does not consider requirement dependencies.	It enables an asynchronous communication amongst distributed end users to reveal their needs.
91.	SNIPR [49]	Importance and dependency	Large			*	It heavily needs professional intervention to implement the prioritisation process.	It handles requirement interdependencies.
92.	Social Network Analysis Technique for RP [66]	Importance	Small	*	*	*	A practical evaluation of the technique's performance has not been implemented to investigate its efficiency.	It considers requirement dependencies in the prioritisation process.
93.	Software Architecture Analysis Method for RP [80]	Quality attributes	NS				It does not provide a quality metric. It does not provide low implementation details.	It specifies areas with high potential complexity.
94.	Software Engineering Risk: Understanding and Management for RP [80], [94], [99], [151]	Cost, risk and benefit	Small	*			It does not address requirement interdependencies and aspects of the customers. It requires the involvement of professional analysts to perform the RP process.	It considers the three major aspects to the organisation. It can be used to prioritise system changes.
95.	Software RP Using Fuzzy Multi-attribute Decision Making [124]	NS	NS			*	The performance of this technique has not been evaluated with industrial projects.	NS
96.	Software RP based on non-functional requirements [35]	Importance	Small	*		*	It does not consider SQP and requirement interdependencies during the prioritisation process.	NS
97.	StakeRare [4]	Importance	Large				It heavily relies on the involvement of experts to execute the RP process, and the quality of the requirements depends on the stakeholders' response. It does not consider requirement interdependencies.	It considers the SQP process. It can work well with a large set of requirements.
98.	Technique for Ordering from Similarity to Ideal Solution to Prioritise the Requirements [26], [115]	Business goals and ease of realisation	NS			*	It cannot update ranks whenever new requirements are extracted. It cannot hierarchically organise requirements.	NS
99.	Theme Screening [87]	Importance	NS			*	NS	NS

TABLE 6. ***(Continued.)*** **Analysis of RP techniques in terms of prioritisation criteria, size of requirement sets, limitations and benefits.**

No	Technique	Prioritisation Criteria	Size of Requirement Set	Limitations				Benefits
				Scalability	Time Consumption	Lack of SQP	Other limitations	
100.	Tool-supported Collaborative RP Process [120]	Budget and effort	Small			*	It does not consider requirement interdependencies. Additional empirical evaluation must be conducted to verify and generalise the technique's effectiveness, especially in prioritising a large set of requirements.	NS
101.	Top Ten [10], [13], [92]	Importance	Small	*		*	It is ambiguous in identifying the priority values of requirements.	It works well with a small set of requirements.
102.	Value-Based RP [26], [63]	Goals of the project, business value	NS		*	*	It requires the participation of project managers and experts in performing prioritisation. It cannot hierarchically organise requirements and exhibits poor prerequisite handling.	It handles requirement interdependencies.
103.	VIRP [19]	Feasibility, modifiability, urgency, traceability, testability, completeness, consistency, understandability and non-redundancy	Medium				It does not categorise the ordered requirements, i.e. classification as most essential, medially essential and less essential. It heavily depends on the participation of experts to perform prioritisation and manually executes SQP.	It considers the SQP process.
104.	Value-oriented Prioritisation VOP [95]	Cost, risk, penalty and business values	Small	*	*	*	It does not address the dependencies amongst the requirements during prioritisation.	It considers the business value in prioritising requirements.
105.	Weighted Criteria Analysis [84], [87]	Importance	NS		*	*	It ignores requirement interdependencies.	NS
106.	Wiegers' Matrix technique [52], [80], [117]	Customer benefit, penalty, cost and risk	NS			*	It heavily relies on the involvement of professional analysts or experts in performing RP. It does not handle requirement interdependencies.	It can integrate various assessment criteria to evaluate requirements.
107.	WGW [149]	Importance	Medium	*			It heavily depends on the involvement of experts to perform prioritisation and it is manually executed. It lacks of considering the dependencies among the requirement during prioritisation process.	NS
108.	WinWin [96]	NS	NS			*	The final consensus, especially when biased stakeholders are participating, is hard to obtain.	NS

NS: Not specified

APPENDIX A

See Table 5.

APPENDIX B

See Table 6.

REFERENCES

[1] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin, Germany: Springer-Verlag, 2010.

[2] S. L. Lim, "Social networks and collaborative filtering for large-scale requirements elicitation," Ph.D. dissertation, School Comput. Sci. Eng., Univ. New South Wales, Sydney, NSW, Australia, 2010.

[3] I. Keshta, M. Niazi, and M. Alshayeb, "Towards implementation of requirements management specific practices (SP1.3 and SP1.4) for Saudi Arabian small and medium sized software development organizations," *IEEE Access*, vol. 5, pp. 24162–24183, 2017.

[4] S. L. Lim and A. Finkelstein, "StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE Trans. Softw. Eng.*, vol. 38, no. 3, pp. 707–735, May/Jun. 2012.

[5] M. I. Babar, M. Ghazali, D. N. A. Jawawi, S. M. Shamsuddin, and N. Ibrahim, "PHandler: An expert system for a scalable software requirements prioritization process," *Knowl.-Based Syst.*, vol. 84, pp. 179–202, Aug. 2015.

[6] P. Achimugu, A. Selamat, and R. Ibrahim, "A clustering based technique for large scale prioritization during requirements elicitation," in *Recent Advances on Soft Computing and Data Mining*, vol. 287, T. Herawan, R. Ghazali, and M. Deris, Eds. Cham, Switzerland: Springer, 2014, pp. 30–39.

[7] A. M. Pitangueira, R. S. P. Maciel, and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature," *J. Syst. Softw.*, vol. 103, pp. 267–280, May 2015.

[8] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *J. Syst. Softw.*, vol. 126, pp. 141–156, Apr. 2017.

[9] R. H. Al-Ta'ani and R. Razali, "Process model for systematic requirements prioritisation process in an agile software development environment based on 5S approach: Empirical study," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, p. 846, 2016.

[10] P. Berander and A. Andrews, "Requirements prioritization," in *Engineering and Managing Software Requirements*. Berlin, Germany: Springer, 2005, pp. 69–94.

[11] S. A. K. Farhan and M. Khan, "A critical analysis of techniques for requirement prioritization and open research issues," *Int. J. Rev. Comput.*, no. 1, pp. 1–11, 2009.

[12] D. Port, A. Olkov, and T. Menzies, "Using simulation to investigate requirements prioritization strategies," in *Proc. 23rd IEEE/ACM Int. Conf. Automat. Softw. Eng. (ASE)*, Sep. 2008, pp. 268–277.

[13] S. Forouzani, R. Ahmad, S. Forouzani, and N. Gazerani, "Design of a teaching framework for software requirement prioritization," in *Proc. 8th Int. Conf. Comput. Technol. Inf. Manage. (ICCM)*, vol. 2, 2012, pp. 787–793.

[14] Y. Li, K.-S. Chin, and X.-G. Luo, "Determining the final priority ratings of customer requirements in product planning by MDBM and BSC," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1243–1255, 2012.

[15] Z. Racheva, M. Daneva, K. Sikkel, R. Wieringa, and A. Herrmann, "Do we know enough about requirements prioritization in agile projects: Insights from a case study," in *Proc. 18th IEEE Int. Requirements Eng. Conf. (RE)*, Sep./Oct. 2010, pp. 147–156.

[16] L. Lehtola, M. Kauppinen, and S. Kujala, "Requirements prioritization challenges in practice," in *Proc. 5th Int Conf. Product-Focused Softw. Process Improvement (PROFES)* (Lecture Notes in Computer Science), vol. 3009. Springer, 2004, pp. 497–508.

[17] A. A. Khan, J. W. Keung, F.-E. Amin, and M. Abdullah-Al-Wadud, "SPIIMM: Toward a model for software process improvement implementation and management in global software development," *IEEE Access*, vol. 5, pp. 13720–13741, 2017.

[18] R. H. AL-Ta'ani and R. Razali, "Prioritizing requirements in agile development: A conceptual framework," *Procedia Technol.*, vol. 11, pp. 733–739, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212017313004064

[19] M. Ramzan, M. Jaffar, and A. A. Shahid, "Value based intelligent requirement prioritization (Virp): Expert driven fuzzy logic based prioritization technique," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 3, pp. 1017–1038, 2011.

[20] P. Achimugu, A. Selamat, and R. Ibrahim, "A preference weights model for prioritizing software requirements," in *Computational Collective Intelligence. Technologies and Applications* (Lecture Notes in Computer Science), vol. 8733, D. Hwang, J. J. Jung, and N. T. Nguyen, Eds. Cham, Switzerland: Springer, 2014, pp. 30–39. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-11289-3_4

[21] J. Karlsson, C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements," *Inf. Softw. Technol.*, vol. 39, nos. 14–15, pp. 939–947, 1998.

[22] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *IEEE Softw.*, vol. 14, no. 5, pp. 67–74, Sep. 1997.

[23] K. A. Khan, "A systematic review of software requirements prioritization," M.S. thesis, Blekinge Inst. Technol., Karlskrona, Sweden, 2006.

[24] G. Kaur and S. Bawa, "A survey of requirement prioritization methods," *Int. J. Eng. Res. Technol.*, vol. 2, no. 5, pp. 958–962, 2013.

[25] M. Pergher and B. Rossi, "Requirements prioritization in software engineering: A systematic mapping study," in *Proc. 3rd Int. Workshop Empirical Requirements Eng.*, 2013, pp. 40–44.

[26] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.

[27] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering version 2.3," EBSE, Keele Univ., Newcastle, U.K., Tech. Rep. EBSE 2007-001, 2007, p. 1051.

[28] R. Razali and F. Anwar, "Selecting the right stakeholders for requirements elicitation: A systematic approach," *J. Theor. Appl. Inf. Technol.*, vol. 33, no. 2, pp. 250–257, 2011.

[29] S. L. Lim, D. Quercia, and A. Finkelstein, "StakeNet: Using social networks to analyse the stakeholders of large-scale software projects," in *Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng. (ICSE)*, vol. 1, May 2010, pp. 295–304.

[30] O. Preiss and A. Wegmann, "Stakeholder discovery and classification based on systems science principles," in *Proc. 2nd Asia–Pacific Conf. Qual. Softw.*, 2001, pp. 194–198.

[31] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, vol. 5, pp. 16173–16192, 2017.

[32] B. A. Kitchenham, E. Mendes, and G. H. Travassos, "Cross versus within-company cost estimation studies: A systematic review," *IEEE Trans. Softw. Eng.*, vol. 33, no. 5, pp. 316–329, May 2007.

[33] C. G. Şen and H. Baraçli, "Fuzzy quality function deployment based methodology for acquiring enterprise software selection requirements," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 3415–3426, 2010.

[34] K. Logue and K. McDaid, "Handling uncertainty in agile requirement prioritization and scheduling using statistical simulation," in *Proc. Agile Conf.*, 2008, pp. 73–82.

[35] U. Garg and A. Singhal, "Software requirement prioritization based on non-functional requirements," *Proc. 7th Int. Conf. Confluence Cloud Comput. Data Sci. Eng.*, vol. 6, Jan. 2017, pp. 793–797.

[36] *CHAOS Report 2015*, Standish Group, Boston, MA, USA, 2015.

[37] M. I. Babar, M. Ramzan, and S. A. K. Ghayyur, "Challenges and future trends in software requirements prioritization," in *Proc. IEEE Int. Conf. Comput. Netw. Inf. Technol. (ICCNIT)*, Jul. 2011, pp. 319–324.

[38] P. Achimugu, A. Selamat, and R. Ibrahim, "Using the fuzzy multi-criteria decision making approach for software requirements prioritization," *J. Teknol.*, vol. 2, no. 13, pp. 21–25, 2015.

[39] F. Hujainah, R. B. A. Bakar, B. Al-Haimi, and M. A. Abdulgabber, "Investigation of stakeholder analysis in requirement prioritization techniques," *Adv. Sci. Lett.*, vol. 24, no. 10, pp. 7227–7231, 2018.

[40] S. L. Lim, D. Damian, and A. Finkelstein, "StakeSource2.0: Using social networks of stakeholders to identify and prioritise requirements," in *Proc. 33rd Int. Conf. Softw. Eng. (ICSE)*, 2011, pp. 1022–1024.

[41] P. Achimugu, A. Selamat, and R. Ibrahim, "A Web-based multi-criteria decision making tool for software requirements prioritization," in *Computational Collective Intelligence. Technologies and Applications* (Lecture Notes in Computer Science), vol. 8733, D. Hwang, J. J. Jung, and N. T. Nguyen, Eds. Cham, Switzerland: Springer, 2014, pp. 444–453. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-11289-3_45

[42] A. Iqbal, "A hybrid technique for requirements," Dept. Comput. Sci. Softw. Eng., Int. Islamic Univ., Islamabad, Pakistan, Tech. Rep., 2012.
[43] Q. Ma, "The effectiveness of requirements prioritization techniques for a medium to large number of requirements: A systematic literature review," M.S. thesis, School Comput. Math. Sci., Auckland Univ. Technol., Auckland, New Zealand, 2009.
[44] F. Creplet, O. Dupouet, F. Kern, B. Mehmanpazir, and F. Munier, "Consultants and experts in management consulting firms," *Res. Policy*, vol. 30, no. 9, pp. 1517–1535, 2001.
[45] M. Almaliki, F. Faniyi, R. Bahsoon, K. Phalp, and R. Ali, "Requirements-driven social adaptation: Expert survey," in *Requirements Engineering: Foundation for Software Quality* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8396. Cham, Switzerland: Springer, 2014, pp. 72–87. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-05843-6_6
[46] P. Achimugu, A. Selamat, and R. Ibrahim, "ReproTizer: A fully implemented software requirements prioritization tool," in *Transactions on Computational Collective Intelligence XXII*. Berlin, Germany: Springer, 2016, pp. 80–105.
[47] F. Hujainah, R. B. A. Bakar, B. Al-Haimi, and A. B. Nasser, "Analyzing requirement prioritization techniques based on the used aspects," *Res. J. Appl. Sci.*, vol. 11, no. 6, pp. 327–332, 2016.
[48] C. Duan, P. Laurent, J. Cleland-Huang, and C. Kwiatkowski, "Towards automated requirements prioritization and triage," *Requirements Eng.*, vol. 14, no. 2, pp. 73–89, 2009.
[49] J. McZara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers—A controlled experiment," *Empirical Softw. Eng.*, vol. 20, no. 6, pp. 1721–1761, 2015.
[50] V. Veerappa, "Clustering methods for requirements selection and optimisation," Ph.D. dissertation, School Comput. Sci. Eng., Univ. College London, London, U.K., Sep. 2012.
[51] S. U. R. Khan, S. P. Lee, M. Dabbagh, M. Tahir, M. Khan, and M. Arif, "RePizer: A framework for prioritization of software requirements," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 8, pp. 750–765, 2016.
[52] H. C. Benestad and J. E. Hannay, "Does the prioritization technique affect stakeholders' selection of essential software product features?" in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, 2012, pp. 261–270.
[53] F. Hujainah, R. B. A. Bakar, B. Al-Haimi, and M. A. Abdulgabber, "Stakeholder quantification and prioritisation research: A systematic literature review," *Inf. Softw. Technol.*, vol. 102, pp. 85–99, Oct. 2018.
[54] P. Voola and A. V. Babu, "Requirements uncertainty prioritization approach: A novel approach for requirements prioritization," *Softw. Eng. Int. J.*, vol. 2, no. 2, pp. 37–49, 2012.
[55] M. Aasem, M. Ramzan, and A. Jaffar, "Analysis and optimization of software requirements prioritization techniques," in *Proc. Int. Conf. Inf. Emerg. Technol. (ICIET)*, 2010, pp. 1–6.
[56] C. K. Kwong and H. Bai, "A fuzzy AHP approach to the determination of importance weights of customer requirements in quality function deployment," *J. Intell. Manuf.*, vol. 13, no. 5, pp. 367–377, Oct. 2002.
[57] K. Y. Chan, C. K. Kwong, and T. S. Dillon, "Determination of importance of customer requirements using the fuzzy AHP method," in *Computational Intelligence Techniques for New Product Design*. New York, NY, USA: Springer, 2012.
[58] M. Vestola, "A comparison of nine basic techniques for requirements prioritization," Helsinki Univ. Technol., Espoo, Finland, Tech. Rep., 2010. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C31&q=A+comparison+of+nine+basic+techniques+for+requirements++prioritization&btnG=
[59] M. Yousuf, M. U. Bokhari, and M. Zeyauddin, "An analysis of software requirements prioritization techniques: A detailed survey," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, 2016, pp. 3966–3970.
[60] D. Greer and G. Ruhe, "Software release planning: An evolutionary and iterative approach," *Inf. Softw. Technol.*, vol. 46, no. 4, pp. 243–253, 2004.
[61] F. Sher, D. N. A. Jawawi, R. Mohamad, and M. I. Babar, "Multi-aspects based requirements priortization technique for value-based software developments," in *Proc. Int. Conf. Emerg. Technol. (ICET)*, 2014, pp. 1–6.
[62] F. M. Kifetew, A. Susi, D. Muñante, A. Perini, A. Siena, and P. Busetta, "Towards multi-decision-maker requirements prioritisation via multiobjective optimisation," in *Proc. 29th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Essen Germany, Jun. 2017, pp. 137–144.
[63] N. Kukreja, S. S. Payyavula, B. Boehm, and S. Padmanabhuni, "Value-based requirements prioritization: Usage experiences," *Procedia Comput. Sci.*, vol. 16, pp. 806–813, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050913000859
[64] C. Li, M. van den Akker, S. Brinkkemper, and G. Diepen, "An integrated approach for requirement selection and scheduling in software release planning," *Requirements Eng.*, vol. 15, no. 4, pp. 375–396, 2010.
[65] A. Sureka, "Requirements prioritization and next-release problem under non-additive value conditions," in *Proc. Austral. Softw. Eng. Conf. (ASWEC)*, 2014, pp. 120–123.
[66] P. Fitsilis, V. Gerogiannis, L. Anthopoulos, and I. K. Savvas, "Supporting the requirements prioritization process using social network analysis techniques," in *Proc. 19th IEEE Int. Workshops Enabling Technol., Infrastructures Collaborative Enterprises*, Jun. 2010, pp. 110–115.
[67] P. Tonella, A. Susi, and F. Palma, "Interactive requirements prioritization using a genetic algorithm," *Inf. Softw. Technol.*, vol. 55, no. 1, pp. 173–187, 2013.
[68] S. I. Majumdar, M. S. Rahman, and M. M. Rahman, "Thorny issues of stakeholder identification and prioritization in requirement engineering process," *IOSR J. Comput. Eng.*, vol. 15, no. 5, pp. 73–78, 2013.
[69] M. Svahnberg, T. Gorschek, R. Feldt, R. Torkar, S. Bin Saleem, and M. U. Shafique, "A systematic review on strategic release planning models," *Inf. Softw. Technol.*, vol. 52, no. 3, pp. 237–248, 2010.
[70] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. N. O. Dag, "An industrial survey of requirements interdependencies in software product release planning," in *Proc. IEEE Int. Conf. Requirements Eng.*, Aug. 2001, pp. 84–91.
[71] R. H. Al-Ta, "ani and R. Razali, "A framework for requirements prioritisation process in an agile software development environment: Empirical study," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, p. 846, 2016.
[72] A. B. M. Moniruzzaman and D. S. A. Hossain. (Jul. 2013). "Comparative study on agile software development methodologies." [Online]. Available: https://arxiv.org/abs/1307.3356
[73] S. Bhatti, V. Tayal, and P. Gulia, "Agile software development," *J. Comput. Sci. Eng.*, vol. 1, no. 4, pp. 108–112, 2015.
[74] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Hum. Behav.*, vol. 51, pp. 915–929, Oct. 2015.
[75] L. Karlsson, M. Höst, and B. Regnell, "Evaluating the practical use of different measurement scales in requirements prioritisation," in *Proc. Int. Symp. Empir. Softw. Eng.*, 2006, pp. 326–335.
[76] C. Li, J. M. van den Akker, S. Brinkkemper, and G. Diepen, "Integrated requirement selection and scheduling for the release planning of a software product," in *Requirements Engineering: Foundation for Software Quality* (Lecture Notes in Computer Science), P. Sawyer, B. Paech, and P. Heymans, Eds. Berlin, Germany: Springer, 2007, pp. 93–108.
[77] H. Bendjenna, P. Charre, and N. E. Zarour, "Using multi-criteria analysis to prioritize stakeholders," *J. Syst. Inf. Technol.*, vol. 14, no. 3, pp. 264–280, 2012.
[78] P. Avesani, C. Bazzanella, A. Perini, and A. Susi, "Supporting the requirements prioritization process. A machine learning approach," in *Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng. (SEKE)*. Banff, AB, Canada: KSI Press, 2004, pp. 306–311.
[79] C. Procte and M. Businge, "Involve users or fail: An IT project case study from East Africa," *Int. J. Inf. Technol. Project Manag.*, vol. 4, no. 4, pp. 51–65, 2013.
[80] R. Thakurta, "A framework for prioritization of quality requirements for inclusion in a software project," *Softw. Qual. J.*, vol. 21, no. 4, pp. 573–597, 2013.
[81] H. Kaiya, H. Horai, and M. Saeki, "AGORA: Attributed goal-oriented requirements analysis method," in *Proc. IEEE Joint Int. Conf. Requirements Eng.*, Sep. 2002, pp. 13–22.
[82] M. Daneva and A. Herrmann, "Requirements prioritization based on benefit and cost prediction: A method classification framework," in *Proc. 34th EUROMICRO Conf. Softw. Eng. Adv. Appl. SEAA EUROMICRO*, 2008, pp. 240–247, 2008.
[83] T. Bebensee, I. van de Weerd, and S. Brinkkemper, "Binary priority list for prioritizing software requirements," in *Requirements Engineering: Foundation for Software Quality* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6182. Berlin, Germany: Springer, 2010, pp. 67–78. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-14192-8_8

[84] Z. Racheva, M. Daneva, and L. Buglione, "Supporting the dynamic reprioritization of requirements in agile development of software products," in *Proc. 2nd Int. Workshop IEEE Softw. Prod. Manag. (IWSPM)*, Sep. 2008, pp. 49–58.

[85] X. Liu, Y. Sun, C. S. Veera, Y. Kyoya, and K. Noguchi, "Priority assessment of software process requirements from multiple perspectives," *J. Syst. Softw.*, vol. 79, no. 11, pp. 1649–1660, 2006.

[86] P. Chatzipetrou, L. Angelis, P. Rovegård, and C. Wohlin, "Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework," in *Proc. 36th EUROMICRO Conf. Softw. Eng. Adv. Appl. (SEAA)*, 2010, pp. 361–370.

[87] Z. Racheva, M. Daneva, A. Herrmann, and R. J. Wieringa, "A conceptual model and process for client-driven agile requirements prioritization," in *Proc. 4th Int. Conf. Res. Challenges Inf. Sci.*, 2010, pp. 287–298.

[88] T. M. Fehlmann, "New Lanchester theory for requirements prioritization," in *Proc. 2nd Int. Workshop Softw. Product Manage. (ISWPM)*, 2008, pp. 35–40.

[89] V. Ahl, "An experimental comparison of five prioritization methods—Investigating ease of use, accuracy and scalability," M.S. thesis, School Eng., Blekinge Inst. Technol., Karlskrona, Sweden, 2005.

[90] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, "Pair-wise comparisons versus planning game partitioning–experiments on requirements prioritisation techniques," *Empirical Softw. Eng.*, vol. 12, no. 1, pp. 3–33, 2007.

[91] G. S. Wasserman, "On how to prioritize design requirements during the QFD planning process," *IIE Trans.*, vol. 25, no. 3, pp. 59–65, 1993.

[92] J. Karlsson, "Software requirements prioritizing," in *Proc. 2nd Int. Conf. Requirements Eng.*, 1996, pp. 110–116.

[93] P. Voola and A. V. Babu, "Interval evidential reasoning algorithm for requirements prioritization," in *Proc. InConINDIA AISC*, vol. 132, 2012, pp. 915–922.

[94] V. Gaur, A. Soni, and P. Bedi, "An agent-oriented approach to requirements engineering," in *Proc. IEEE 2nd Int. Advance Comput. Conf. (IACC)*, Feb. 2010, pp. 449–454.

[95] J. Azar, R. K. Smith, and D. Cordes, "Value-oriented requirements prioritization in a small development organization," *IEEE Softw.*, vol. 24, no. 1, pp. 32–37, Jan./Feb. 2007.

[96] P. Gruenbacher, "Collaborative requirements negotiation with EasyWin-Win," *Proc. 2nd Int. Workshop Requireds Eng. Process.*, Greenwich, U.K., 2000, pp. 954–958.

[97] M. A. Iqbal, A. M. Zaidi, and S. Murtaza, "A new requirement prioritization model for market driven products using analytical hierarchical process," in *Proc. Int. Conf. Data Storage Data Eng. (DSDE)*, 2010, pp. 142–149.

[98] A. Perini, A. Susi, and P. Avesani, "A machine learning approach to software requirements prioritization," *IEEE Trans. Softw. Eng.*, vol. 39, no. 4, pp. 445–461, Apr. 2013.

[99] D. Greer, D. W. Bustard, and T. Sunazuka, "Prioritisation of system changes using cost-benefit and risk assessments," in *Proc. IEEE Int. Symp. Requirements Eng.*, Jun. 1999, pp. 180–187.

[100] H. P. In, D. Olson, and T. Rodgers, "Multi-criteria preference analysis for systematic requirements negotiation," in *Proc. IEEE Comput. Soc. Int. Comput. Softw. Appl. Conf.*, Aug. 2002, pp. 887–892.

[101] P. Berander, "Prioritization of stakeholder needs in Software engineering: Understanding and evaluation," Ph.D. dissertation, Blekinge Inst. Technol., Karlskrona, Sweden, 2004.

[102] P. Avesani, C. Bazzanella, A. Perini, and A. Susi, "Facing scalability issues in requirements prioritization with machine learning techniques," in *Proc. 13th IEEE Int. Conf. Requirements Eng.*, Aug./Sep. 2005, pp. 297–305.

[103] B. Patrik and P. Jönsson, "Hierarchical cumulative voting (HCV)—Prioritization of requirements in hierarchies," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 16, no. 6, pp. 819–849, 2006.

[104] S. Hatton, "Choosing the right prioritisation method," in *Proc. Aust. Softw. Eng. Conf. (ASWEC)*, 2008, pp. 517–526.

[105] M. R. Beg, R. P. Verma, and A. Joshi, "Reduction in number of comparisons for requirement prioritization using B-Tree," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, Mar. 2009, pp. 340–344.

[106] P. Berander and M. Svahnberg, "Evaluating two ways of calculating priorities in requirements hierarchies—An experiment on hierarchical cumulative voting," *J. Syst. Softw.*, vol. 82, no. 5, pp. 836–850, 2009.

[107] M. Sadiq and M. Shahid, "Elicitation and prioritization of software requirements," *Int. J. Recent Trends Eng.*, vol. 2, no. 3, pp. 138–142, 2009.

[108] M. Svahnberg and A. Karasira, "A study on the importance of order in requirements prioritisation," in *Proc. 3rd Int. Workshop Softw. Product Manage. (IWSPM)*, 2009, pp. 15–22.

[109] N. M. Carod and A. Cechich, "Cognitive-driven requirements prioritization: A case study," in *Proc. 9th IEEE Int. Conf. Cognit. Inform. (ICCI)*, Jul. 2010, pp. 75–82.

[110] D. C. Lima, F. Freitas, G. Campos, and J. Souza, "A fuzzy approach to requirements prioritization," in *Search Based Software Engineering* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2011, pp. 64–69.

[111] N. Kukreja, B. Boehm, S. S. Payyavula, and S. Padmanabhuni, "Selecting an appropriate framework for value-based requirements prioritization," in *Proc. 20th IEEE Int. Requirements Eng. Conf. (RE)*, Sep. 2012, pp. 303–308.

[112] M. Khari and N. Kumar, "Comparison of six prioritization techniques for software requirements," *J. Global Res. Comput. Sci.*, vol. 4, no. 1, pp. 38–43, 2013.

[113] P. Bajaj and V. Arora, "A novel approach to select an appropriate requirements prioritization technique," *Int. J. Comput. Appl.*, vol. 77, no. 9, pp. 25–30, 2013.

[114] V. A. Babu and P. Voola, "4A frameworks for requirements prioritization," *Int. J. Comput. Appl.*, vol. 76, no. 1, pp. 38–44, 2013.

[115] N. Kukreja, "Decision theoretic requirements prioritization: A two-step approach for sliding towards value realization," in *Proc. Int. Conf. Softw. Eng.*, 2013, pp. 1465–1467.

[116] C.-C. Huang, F.-Y. Lin, F. Y.-S. Lin, and Y. S. Sun, "A novel approach to evaluate software vulnerability prioritization," *J. Syst. Softw.*, vol. 86, no. 11, pp. 2822–2840, 2013.

[117] K. Wiegers and J. Beatty, "First things first: Setting requirement priorities," in *Software Requirements*, 3rd ed. Redmond, WA, USA: Microsoft Press, 2013.

[118] R. Easmin, A. Ul Gias, and S. M. Khaled, "A partial order assimilation approach for software requirements prioritization," in *Proc. Int. Conf. Inform., Electron. Vis. (ICIEV)*, 2014, pp. 1–5.

[119] M. Sadiq, T. Hassan, and S. Nazneen, "AHP_GORE_PSR: Applying analytic hierarchy process in goal oriented requirements elicitation method for the prioritization of software requirements," in *Proc. 3rd IEEE Int. Conf.*, Feb. 2017, pp. 1–5.

[120] P. Busetta, F. M. Kifetew, D. Munante, A. Perini, A. Siena, and A. Susi, "Tool-supported collaborative requirements prioritisation," in *Proc. Int. Comput. Softw. Appl. Conf.*, vol. 1, 2017, pp. 180–189.

[121] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "An adaptive fuzzy decision matrix model for software requirements prioritization," in *Advanced Approaches to Intelligent Information and Database Systems*, N. T. Nguyen and R. Kowalczyk, Eds. Cham, Switzerland: Springer, 2014, pp. 129–138.

[122] P. Achimugu and A. Selamat, "A hybridized approach for prioritizing software requirements based on K-means and evolutionary algorithms," in *Computational Intelligence Applications in Modeling and Control*. Cham, Switzerland: Springer, 2015, pp. 73–93.

[123] B. B. Jawale, G. K. Patnaik, and A. T. Bhole, "Requirement prioritization using adaptive fuzzy hierarchical cumulative voting," in *Proc. 7th IEEE Int. Adv. Comput. Conf. (IACC)*, Jan. 2017, pp. 95–102.

[124] A. Ejnioui, C. E. Otero, and A. A. Qureshi, "Software requirement prioritization using fuzzy multi-attribute decision making," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Oct. 2012, pp. 1–6.

[125] B. Regnell, L. Karlsson, and M. Höst, "An analytical model for requirements selection quality evaluation in product software development," in *Proc. IEEE Int. Conf. Requirements Eng.*, Sep. 2003, pp. 254–263.

[126] N. Garg, M. Sadiq, and P. Agarwal, "GOASREP: Goal oriented approach for software requirements elicitation and prioritization using analytic hierarchy," in *Proc. 5th Int. Conf. Frontiers Intell. Comput., Theory Appl.*, vol. 516, 2017, pp. 281–287.

[127] R. V. Anand and M. Dinakaran, "Handling stakeholder conflict by agile requirement prioritization using Apriori technique," *Comput. Elect. Eng.*, vol. 61, pp. 126–136, Jul. 2017.

[128] I. Morales-Ramirez, D. Muñante, F. Kifetew, A. Perini, A. Susi, and A. Siena, "Exploiting user feedback in tool-supported multi-criteria requirements prioritization," in *Proc. IEEE Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Sep. 2017, pp. 424–429.

[129] K. S. Ahmad, N. Ahmad, H. Tahir, and S. Khan, "Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements," in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, 2017, pp. 433–437.

[130] M. S. Rahim, A. Z. M. E. Chowdhury, and S. Das, ''Rize: A proposed requirements prioritization technique for agile development,'' in *Proc. IEEE Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Dec. 2017, pp. 634–637.

[131] D. Mougouei and D. M. W. Powers, ''Modeling and selection of interdependent software requirements using fuzzy graphs,'' *Int. J. Fuzzy Syst.*, vol. 19, no. 6, pp. 1812–1828, 2017.

[132] M. Sadiq and S. K. Jain, ''Applying fuzzy preference relation for requirements prioritization in goal oriented requirements elicitation process,'' *Int. J. Syst. Assurance Eng. Manage.*, vol. 5, no. 4, pp. 711–723, 2014.

[133] N. Seyff, I. Todoran, K. Caluser, L. Singer, and M. Glinz, ''Using popular social network sites to support requirements elicitation, prioritization and negotiation,'' *J. Internet Services Appl.*, vol. 6, no. 1, pp. 1–16, 2015.

[134] S. A. Asif, Z. Masud, R. Easmin, and A. Ul Gias, ''SAFFRON: A semi-automated framework for software requirements prioritization,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 12, pp. 491–499, 2017.

[135] J. del Sagrado, I. M. del Águila, and F. J. Orellana, ''Multi-objective ant colony optimization for requirements selection,'' *Empirical Softw. Eng.*, vol. 20, no. 3, pp. 577–610, 2015.

[136] M. Dabbagh and S. P. Lee, ''An approach for integrating the prioritization of functional and nonfunctional requirements,'' *Sci. World J.*, vol. 2014, Apr. 2014, Art. no. 737626.

[137] N. Sharif, K. Zafar, and W. Zyad, ''Optimization of requirement prioritization using computational intelligence technique,'' in *Proc. Int. Conf. Robot. Emerg. Allied Technol. Eng. (iCREATE)*, 2014, pp. 228–234.

[138] G. Ruhe, A. Eberlein, and D. Pfahl, ''Quantitative WinWin: A new method for decision support in requirements negotiation,'' *Proc. 14th Int. Conf. Softw. Eng. Knowl. Eng.*, May 2014, pp. 159–166.

[139] G. Ruhe, A. Eberlein, and D. Pfahl, ''Trade-off analysis for requirements selection,'' *Int. J. Softw. Eng. Knowl. Eng.*, vol. 13, no. 4, pp. 345–366, 2003.

[140] Y. Zhang, M. Harman, and S. A. Mansouri, ''The multi-objective next release problem,'' in *Proc. ACM Genet. Evol. Comput. Conf.*, 2007, pp. 1129–1136.

[141] A. Alzaqebah, R. Masadeh, and A. Hudaib, ''Whale optimization algorithm for requirements prioritization,'' in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jan. 2018, pp. 84–89.

[142] N. Misaghian and H. Motameni, ''An approach for requirements prioritization based on tensor decomposition,'' *Requirements Eng.*, vol. 23, no. 2, pp. 169–188, 2018.

[143] L. Alawneh, ''Requirements prioritization using hierarchical dependencies,'' in *Information Technology—New Generations*. Cham, Switzerland: Springer, 2018, pp. 459–464.

[144] I. Ibriwesh, S.-B. Ho, and I. Chai, ''Overcoming scalability issues in analytic hierarchy process with ReDCCahp: An empirical investigation,'' *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 7995–8011, 2018.

[145] N. Condori-Fernandez, M. F. Granda, and T. E. J. Vos, ''Towards a functional requirements prioritization with early mutation testing,'' in *Proc. 5th Int. Workshop Requirements Eng. Test. (RET)*, Apr. 2018, pp. 21–24.

[146] Y. V. Singh, B. Kumar, S. Chand, and J. Kumar, "A comparative analysis and proposing 'ANN fuzzy AHP model' for requirements prioritization," *Int. J. Inf. Technol. Comput. Sci.*, vol. 10, no. 4, pp. 55–65, 2018.

[147] Á. García-Soler, D. Facal, U. Díaz-Orueta, L. Pigini, L. Blasi, and R. Qiu, ''Inclusion of service robots in the daily lives of frail older users: A step-by-step definition procedure on users' requirements,'' *Arch. Gerontol. Geriatrics*, vol. 74, pp. 191–196, Jan. 2018.

[148] M. S. Krishnan, ''RFP based requirement prioritization—A one-step solution,'' *Mater. Today Proc.*, vol. 5, no. 1, pp. 642–649, 2018.

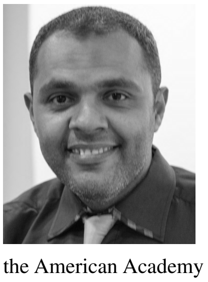
[149] A. Hudaib, R. Masadeh, and A. Alzaqebah, ''WGW: A hybrid approach based on whale and grey wolf optimization algorithms for requirements prioritization,'' *Adv. Syst. Sci. Appl.*, vol. 2, no. 2, pp. 63–83, Aug. 2018.

[150] H. Ahuja, Sujata, and U. Batra, ''Performance enhancement in requirement prioritization by using least-squares-based random genetic algorithm,'' in *Proc. Best Sel. Papers 3rd Int. Conf. REDSET Innov. Comput. Intell.*, B. Panda, S. Sharma, and U. Batra, Eds. Singapore: Springer, 2018, pp. 251–263.

[151] P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*. Boston, MA, USA: Addison-Wesley, 2002.

FADHL HUJAINAH received the B.Sc. degree (Hons.) in computer science and software engineering and the M.Sc. degree (Hons.) from the University of Technology at Malaysia, Johor Bahru, Malaysia, in 2012 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Kuantan, Malaysia. His research interests include software engineering with particular interest in requirements' prioritization and stakeholder analysis.

ROHANI BINTI ABU BAKAR received the B.Sc. degree in computer science from the University of Technology at Malaysia, Johor Bahru, Malaysia, the M.Sc. degree in computer science from the Universiti Malaya, Kuala Lumpur, Malaysia, and the Dr.Eng. degree in DNA computing and its applications from Waseda University, Tokyo, Japan. She is a Senior Lecturer with the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Kuantan, Malaysia. Her research interests include software engineering and soft computing.



MANSOOR ABDULLATEEF ABDULGABBER received the B.C.A. degree (First Division) from Osmania University, Hyderabad, India, the M.Sc. degree in real-time software engineering from the Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia, and the Ph.D. degree in software testing and semantic technology from the Universiti Malaya, Kuala Lumpur, Malaysia. He is a specialist in the area of software engineering and a certified Project Manager, as well as a fellow with the American Academy of Project Management. He has been in the industry for over 10 years. He is currently an Assistant Professor with the University of Prince Mugrin, Madinah, Saudi Arabia. He is highly professional, passionate, and focused in what he does and wishes to accomplish. His research interests include software engineering.

KAMAL Z. ZAMLI (M'17) received the B.Sc. degree in electrical engineering from the Worcester Polytechnic Institute, USA, in 1992, the M.Sc. degree in real-time software engineering from the Universiti Teknologi Malaysia in 2000, and the Ph.D. degree in software engineering from Newcastle University, Newcastle upon Tyne, U.K., in 2003. He is currently the Dean and a Professor with the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang.

His main research interests include search-based software engineering, combinatorial software testing, and computational intelligence. He is a member of MySEIG.