Computer Standards & Interfaces 69 (2020) 103389

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A systematic literature review on requirement prioritization techniques and their empirical evaluation

Faiza Allah Bukhsh[a], Zaharah Allah Bukhsh[b], Maya Daneva[*,a]

[a] Department of Computer Science, University of Twente, Enschede, the Netherlands
[b] Department of Construction Management and Engineering, University of Twente, Enschede, the Netherlands

ARTICLE INFO

Keywords:
Requirements engineering
Requirements prioritization
Empirical study
Empirical research method
Systematic literature review

ABSTRACT

[Context and Motivation] Many requirements prioritization approaches have been proposed, however not all of them have been investigated empirically in real-life settings. As a result, our knowledge of their applicability and actual use is incomplete. **[Question/problem]** A 2007 systematic review on requirements prioritization mapped out the landscape of proposed prioritization approaches and their prioritization criteria. To understand how this sub-field of requirements engineering has developed since 2007 and what evidence has been accumulated through empirical evaluations, we carried out a literature review that takes as input publications published between 2007 and 2019. **[Principle ideas/results]** We evaluated 102 papers that proposed and/or evaluated requirements prioritization methods. Our results show that the newly proposed requirements prioritization methods tend to use as basis fuzzy logic and machine learning algorithms. We also concluded that the Analytical Hierarchy Process is the most accurate and extensively used requirement prioritization method in industry. However, scalability is still its major limitation when requirements are large in number. We have found that machine learning has shown potential to deal with this limitation. Last, we found that experiments were the most used research method to evaluate the various aspects of the proposed prioritization approaches. **[Contribution]** This paper identified and evaluated requirements prioritization techniques proposed between 2007 and 2019, and derived some trends. Limitations of the proposals and implications for research and practice are identified as well.

1. Introduction

The concept of Requirement Prioritization (RP) emerged with the increased demand of complex software/information systems by stakeholders [20]. RP is a decision making task in which software engineers collaborate with stakeholders, to understand their demands, in order to define the implementation order of requirements by considering the budget, time and technical constraints. More often than not, the increasing number of requirements and the growing pool of stakeholders with diverse interests make the RP task even harder. Numerous techniques have been introduced in the literature to make the RP task accurate, efficient, reliable and conflict-free. However, each technique comes with its limitations and makes explicit and implicit assumptions about the project context where RP takes place [5]. These assumptions need to be taken into consideration when a RP method is evaluated empirically, be it for usefulness, utility, applicability or effectiveness. In this systematic literature review (SLR) we want to evaluate the empirical evidence that has been generated by empirical studies in RP. Our work follows up on a previously done SLR [53], which covered the relevant literature on RP up to the year 2007. In the present SLR, we focus on studies that are published between 2007 and 2019. Analyzing the publications in the past 12 years is useful for identifying trends and revising our knowledge about the empirical evaluation and validation efforts going on in the RP sub-area of Requirement Engineering. Moreover, this work complements the 2014 SLR of Achimugu et al. [4] which outlined the most recently proposed requirement prioritization approaches along with their limitations. However, this 2014 review paid little attention to those studies that evaluate the proposed requirement prioritization techniques. In contrast to these authors' work, here we performed a meta-analysis of empirical evaluation studies in addition to studies that are about proposals of RP methods. Our classification of literature is inspired by the publication of Wieringa et al. [117], which clearly articulates the differences between validation/evaluation research and solution proposal research. The purpose of our SLR is, thus, twofold: a) to assess and analyze the RP techniques, algorithms, methods, and approaches published from 2007 to 2019 and

* Corresponding author.
E-mail addresses: f.a.bukhsh@utwente.nl (F.A. Bukhsh), z.allahbukhsh@utwente.nl (Z.A. Bukhsh), m.daneva@utwente.nl (M. Daneva).

https://doi.org/10.1016/j.csi.2019.103389
Received 12 March 2019; Received in revised form 16 August 2019; Accepted 6 November 2019
Available online 13 November 2019
0920-5489/

b) to consolidate the empirical evidence provided in empirical studies that validate the effectiveness of these individual approaches through comparisons, experiments, case studies and surveys.

The rest of the paper is structured as follows: Section 2 presents our research questions, search strategy, study selection criteria, and the execution of our research process. Section 3 reports on our findings to the research questions and presents the proposed RP methods and validation/evaluation studies since 2007. Section 4 discusses the findings. Section 5 is on validity threats. Section 6 explains how our findings complement previously published results in related SLR and empirical studies of other authors. Section 7 presents implications for research and practice. Finally Section 8 concludes.

2. Research methods

For the purpose of our review, we used the guidelines of Kitchenham et al. [64]. This section provides a complete overview of how we applied them, from formulating research questions to the critical appraisal of collected studies. The following sub-sections describe our search strategy, search strings, inclusion and exclusion criteria and quality assessment of included studies which ensures our rigorous approach for this SLR.

2.1. Research questions

We formulated three Research Questions (RQs) to pursue the critical analysis of recently conducted research in RP methods and RP method evaluation: **RQ1:** *What are the published software RP methods/techniques/approaches in the last decade (2007 to 2019)?* **RQ2:** *What empirical evidence has been produced in the scientific literature about RP methods/techniques/approaches that were published after 2007?* **RQ3:** *What evaluation approaches have been used in empirical studies to validate the proposed RP methods/approaches/techniques?*

The population of our RQs are published studies from Sept 2007 to June 2019 and includes relevant empirical studies and publications proposing new prioritization methods, approaches and techniques.

2.2. Search strategy

We followed our three RQs to design our review process which was inspired by the guidelines of Kitchenham et al. [64]. We selected Scopus as the digital library to use because it contains publications from major journals and conference proceedings, which helped us get a diverse set of publications on the subject of RP. Also, we chose it because recent bibliographic research [51] and [81] indicated it as the most comprehensive and user-friendly database. The search in Scopus was done on June 15, 2019, and used the following search string in the article title, abstract, or keywords:

"requirement AND prioritization AND (method OR technique OR approach)"

The possible combinations of keywords relevant to our RQs are helpful to collect those studies that can be used to answer our RQs. Boolean AND and OR conjunctions with keywords are used in digital libraries instead of plain search strings. Moreover, we have used the possible keywords to ensure that none of the relevant material is left out. This string returned 506 papers. We have applied the following restrictions to define the boundaries of our study: (i) limit by source type (i.e. conference papers and journal articles), (ii) limit by publication year, starting from Sept 2007, and (iii) limit by Scopus' subject area, i.e. Computer Science, Engineering or Business. This search was complemented with a second search in other four digital libraries: IEEE Explorer, ACM, Springer, ScienceDirect, and the World of Science to also include materials that could possibly be not within our set of Scopus. This brought 20 more papers, which yielded 526 publications in total. We make the notes that (1) we chose 2007 as the starting year in our search for RP papers, because the previously published research [53] already analysed literature on RP published until 2007, and we wanted to focus on trends that might have formed since then; (2) the composition of our search strings was the result of a learning process [67] including experimentation with a variety of combinations of keywords in order to test synonyms used in literature and to cover the variety of empirical studies on RP techniques; and (3) as part of our search process, we checked the reference lists of the collected studies in order to find any studies that we could have possibly missed. However, this snowballing" exercise as it is called by research methodologists [64], did not result in adding new studies. We found that the reference lists that we checked, correlated with the studies that we have already found, which validated our literature collection approach and assured us that our search string was performing sufficiently well. In order not to miss the latest research this search process executed iteratively from Feb 28, 2019, to June 15, 2019. Moreover, the authors are experts in the field of RP and active in the RE-related conferences, so they are well informed of new studies; the RP topic is specific to the RE field, and it is unlikely that journals in other domains would include publications on this topic.

2.2.1. Inclusion and exclusion criteria

Initially, the studies were selected based on their titles. After removing duplicate papers, we used the following inclusion criteria (**ICs**) and exclusion criteria (**ECs**). The inclusion criteria are:

- IC1 The paper directly relates to the topic of our review. This means, we include papers that explicitly propose a new approach or improvement in an existing approach for RP. In addition, we include papers that judge the effectiveness of RP methods by means of comparative studies, case studies, experiments.
- IC2 The paper addresses the research questions.
- IC3 The paper is published in a peer-reviewed journal, conference or a workshop.
- IC4 The paper is in English.
- IC5 The paper is available for download.

The exclusion criteria are:

- EC1 The paper talks about RP either as main topic or as a side topic that is related to the paper's central topic.
- EC2 The paper performs prioritization based on stakeholder preferences, only. However, studies that initiate prioritization with stakeholders' preferences and then perform further evaluation are included.
- EC3 The paper is not peer-reviewed.

We applied these criteria while reading the abstracts of collected studies. In this process, we categorized the selected studies into two groups: a) Those studies that essentially propose a new prioritization method b) Those empirical studies that either validate the proposed method by experiments, case studies, interviews, surveys, or compare a method with other methods.

2.3. Critical appraisal of collected studies

The purpose of the critical appraisal of our gathered literature is to make a remark on quality of initially collected studies which will be further used to articulate the answers to our three RQs. To assess the quality of each study, the following quality assessment questions with respect to RQs are formulated and applied on each individual study. (For detailed quality scores see Appendix A.)

For ***RQ1***,

- Does the paper propose a new RP method/approach/technique clearly?
- Is the proposed approach based on previously defined methods?

- Is the proposed method empirically evaluated or validated (e.g. by using a realistic example, a case study in a real-life setting, an experiment or another empirical research method)?

For ***RQ2*** & ***RQ3***,

- Does the concluded result in the selected empirical paper match the purpose of the empirical study presented in that paper, w.r.t ***RQ2***?
- Does the empirical study explicitly state its evaluation method w.r.t ***RQ2*** and ***RQ3***?
- As a result of comparison, is the claim supported by a specific validation method w.r.t ***RQ2*** and ***RQ3***?
- Is the purpose of the empirical study clearly defined w.r.t to ***RQ2***?

These questions are formulated by performing careful analysis of our collected literature sources as well as by considering the checklist of Wieringa et al for evaluating the quality of empirical research in Software Engineering and in Information Systems [116] and the guidelines in [64]. These methodologists state that the evaluation or validation technique of a study should be considered as the most important factor to judge its quality. For the critical appraisal of each study, we use the ordinal range of ratings from 0 to 4, where each of our quality assessment questions holds 1 point. A study that attains a value of less than 2, was excluded. This scoring is independent, for example, if a paper scores 2 for RQ1 and doesn't score any point for RQ2&3, it is included. Quality scores of most recent papers are shown in Appendix A.

2.4. Execution of the data extraction process and synthesis strategy

Fig. 1 gives a detailed overview of our research process that we used to gradually exclude studies. It shows that 102 studies in total passed the quality assessment procedure by attaining at least 2 points. The first author has then read the abstracts of all the 526 papers to apply the inclusion and exclusion criteria and assign them a point. The second and third authors double-checked the process by reviewing the removed papers separately to ensure that no papers were mistakenly removed. In the end, the number of papers remained intact.

Fig. 2 shows the distribution of the papers per year of publication. In this figure, we can see that extensive research has been undertaken on the RP topic from the year 2015 till present. Moreover, there is extensive research on the topic of RP in Asia as can be seen in Fig. 6 (for details see Section 3.4).

Once the 102 relevant studies were selected, the following data extraction strategy was used. For ***RQ1***, our data extraction strategy included the identification of prioritization processes, specific frameworks, tools and adaptations of decision-making theories to perform prioritization. Regarding those studies that consisted of a series of publications, data was extracted from the most recently published study in which the proposed RP method was evaluated repeatedly regarding the same evaluation criteria as in the previous studies. In doing so, we assumed that the most recent publication is an improvement and/or an extension of previously published empirical evaluation work. For ***RQ2***, the specific methods, their limitations and the evidence that the application of the method produced, were analyzed and categorized. For the ***RQ3***, the validation and evaluation technique of each study was identified. A short overview is provided in Appendix B.

Next, our data synthesis strategy was qualitative and theoretical in nature [100] and included the following: In each paper, we first searched for the most important messages, findings or lessons learned. These are usually in the answers to the research questions posed in the respective study or are summarized in the conclusion section of the paper. After discerning the learning and messages of all 102 papers, we grouped them in clusters. This helped us also form categories of studies.

3. Results

This section presents our findings for the RQs defined in Section 2.1. Section 3.1 explores studies that proposed new RP methods or techniques. Section 3.2 provides the answer to ***RQ2*** by summarizing the limitations and issues regarding the evidence of using RP methods. Section 3.3 identifies and discusses the validation and evaluation approaches used by those selected studies in our set of 102 that validated RP methods. Finally, Section 3.4. Outlines the demograhic details of the selected studies.

3.1. requirements prioritization approaches (*RQ1*)

Among the 102 papers that presented RP techniques, there are 54 that we used to answer ***RQ1***. Some techniques that were published in these 54 papers were grounded on well-established RP concepts that have been in use for more than 30 years. Those are techniques proposed long before 2007 and the research community kept modifying, extending or expanding them in the period 2007–2019. We found that the most prominent 12 RP techniques that were used as foundations for the methods proposed in our collected studies, are: the Analytical Hierarchical Process (AHP), the Numerical Assignment (NA), the Binary Search Tree (BST),the Planning Game (PG), the Cumulative Voting (CV), the Quality Functional Deployment (QFD), the cost-value approach, the Win-Win method, MosCow, the Top10, the bubble sort approach and the Wieger's approach. These techniques inspired RP researchers in the creation of new RP methods that were published in the period of 2007–2019. In fact, among the 54 selected studies in this section, we found that more than half of the new methods proposed since 2007, are based on the previously existing methods such as AHP, NA, CV and the others from the above list.

As shown in Fig. 3, AHP is the most extensively used method due to its reliability and accuracy [16,27,58,59,87,112,120]. The techniques based on fuzzy logic, stakeholders preferences and advanced data processing are ranked second. In addition, the cost-value approach and the Binary Tree method are also being relatively more frequently used for the RP due to their ability to deal with emerging requirements of each increment efficiently. Next, we notice that according to the published studies, the techniques based on fuzzy logic provide accuracy (e.g. [35,71,83,96,97]), while data-driven techniques are efficient not only in use but also deal with large amount of data [43]. Furthermore, despite the extensive usage of RP methods such as the AHP, a number of evaluation studies [59,70] point out the limitations of these methods. We make the note that our set of 54 papers (see Fig. 3) that describe RP methods and techniques, do not depend solely on those popular methods (e.g. AHP, NA or CV), but suggest improvements of those methods or use one or more ideas from the basic method in a particular step or steps of the RP process.

We have also noticed that more than 25% of the examined 54 papers are based on the non-traditional, emerging technologies such as fuzzy logic, machine learning, data mining, sample selections and group recommendations to prioritize the software requirement in order to deal with their uncertain nature. These findings are discussed into the following sub-section,

3.1.1. Requirements prioritization approaches based on AHP

AHP is one of multi criteria decision making to derive ratio scales for consistant judgment. We have found 15 studies that proposed a new approach based on AHP and pairwise comparison concepts. These studies suggested an extension or modification in AHP, by proposing a model, an algorithm, a multi-layered and hybrid approach for prioritization of functional and quality (or also known as non-functional) requirements. In the following, the brief description of each study is provided.

- [16] proposed a Stratified-AHP (S-AHP) method, which is an

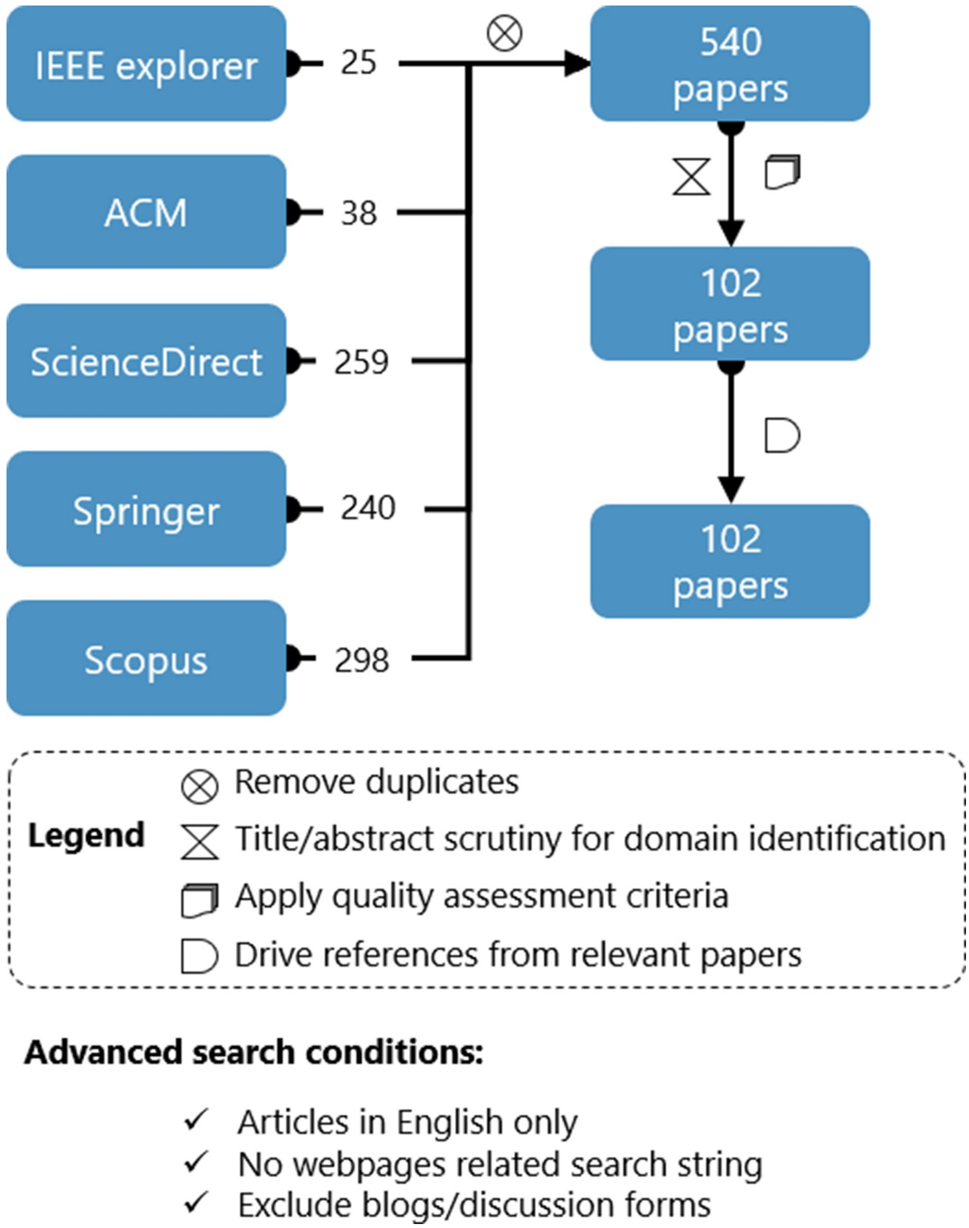

Fig. 1. Study selection process.

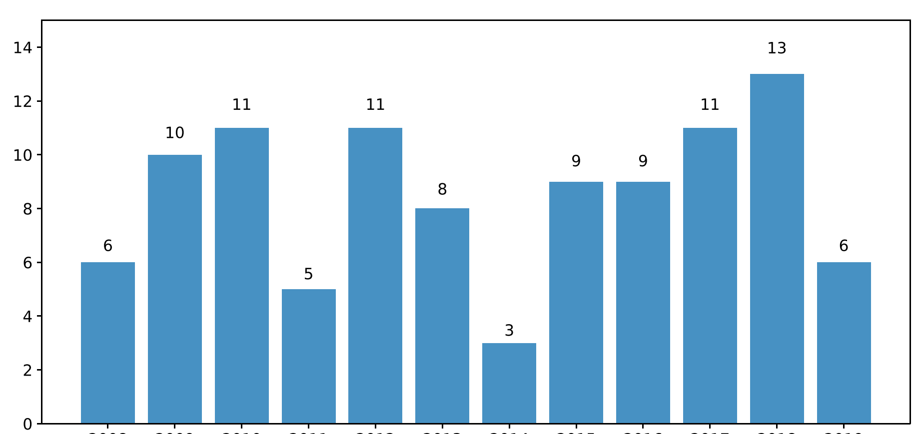

Fig. 2. Our selected studies with respect to year of publication.

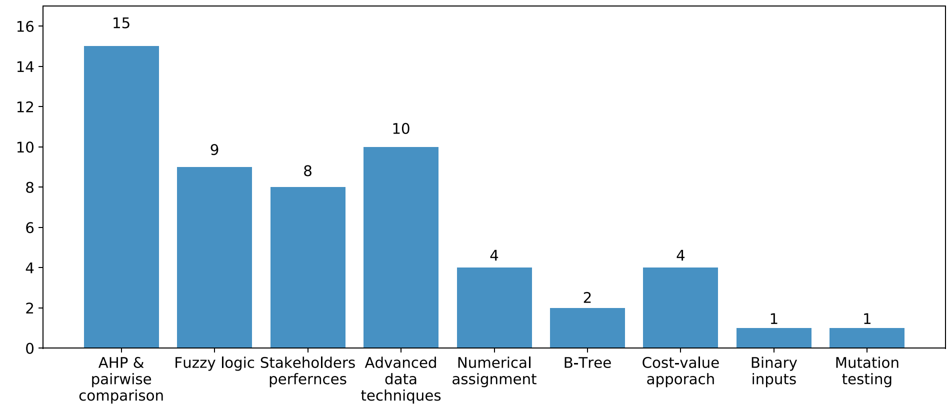

Fig. 3. Categorization of the 54 RP studies.

extension of the AHP. S-AHP prioritizes the software's features based on its qualifiers and concerns in feature models, which are further used in the staged configuration software line. S-AHP ranks and prioritizes the business objectives and its relevant high level goals. Once the stakeholders understand software behaviour against the business objectives, the prioritization of software features' concerns, annotations and qualifiers is performed. This layered approach of pair-wise comparison reduces the number of comparisons tremendously.

- [58] proposed the so-called Market Driven Requirement Prioritization Model (MDRPM), in which AHP is modified in order to be used even with a great number of requirements by reducing the total number of comparisons. MDRPM sets the assumptions of conflict-free and dependency-free pool of requirements. On the basis of 'best-fit' techniques, MDRPM divides the requirements into bins. In the modified AHP engine, first each bin's requirements are prioritized, while as the last step, all the bins are prioritized based on mathematical formulations of AHP. Grouping the requirements into bins reduces the number of comparisons, which makes AHP a scalable prioritization technique for large projects.
- [120] proposed a two-step hybrid method that prioritizes requirements by AHP and importance-based node ranking. Before prioritizing the requirements, this method suggests the identification of requirements dependencies. A network model is developed based on the importance and relative dependencies of requirements. Instead of using the technical tools, this method considers the XML based format for development of network model. Due to its hybrid approach, once the requirements are prioritized based on importance, it is further prioritized by utilizing the AHP traditional method.
- [112] proposed an interactive genetic algorithm in which dependencies among requirements are considered before the prioritization. An ordered set of requirements are listed by the genetic algorithm by reducing the conflicts among requirements. If an interactive genetic algorithm encounter ties state in pairs of requirements, the users are asked to interact and provide its preference values. Once the conflict-free list of ordered requirements is obtained, the further steps of prioritization are performed by AHP.
- [25] propose a fuzzy AHP, by using fuzzy scales that overcomes the limitation of the traditional AHP that cannot address the vagueness and uncertainty inherent in human judgment. Based on a pairwise comparison technique, the comparison matrices of the fuzzy AHP are built with a set of fuzzy triangular numbers.
- [27] proposed an approach that focuses on prioritization of quality requirement, which is contrary to functional requirements based prioritization. The quality requirements of software are mostly dependant and conflicting with each other. The first ordered list is formalized on the importance defined by users' preferences. With the formation of the ordered list, an iterative AHP is performed in which the highest value quality attribute is omitted with each iteration. Prioritization of quality requirements based on user's preferences and AHP yields a highly reliable software system that matches with stakeholder's expectations.
- [50] proposed a multi-layered dynamic approach for re-prioritization of requirements in agile software development. This approach considers the re-prioritization of implement requirements, new requirements and delayed requirements. As a result of change in decision-making process, the re-allotments of rank and prioritization values to implemented and new requirements marks this approach as highly dynamic. This approach does not fully use the AHP due to its dynamic nature but considers the non pairwise comparison methods. However, due to the reliability and accuracy of AHP, this dynamic approach plans to utilize it in the future.
- [40] proposed an AHP-based requirement prioritization approach by considering the interdependence relationship between NFR and FR. This approach high lighted the weakness of AHP during and after a change in existing requirements or introduction of new requirements.
- [59] proposed an AHP-based technique to rank the quality requirements of a software by taking into account the design tactics and design principles. This techniques enable the trade-off analysis of design tactics and design principles for the implementation of software quality requirements.
- [63] proposed an AHP-based prioritization approach, RePizer, that provides a holistic view of the entire project to ensure the judicious of the software development process. The overview of the softwere development process by RePizer is based on the RePizer in terms of expected accuracy and ease of use, it is also considered the guidelines of planning game (PG).
- [59] discuss the AHP technique in order to quantitatively rank design decisions and tactics of software requirement prioritization process. Interdependences between system quality requirements and design tactics and principles are also considered. The proposed approach is evaluated on the remote monitoring system for medical patients. The approach is proved to be helpful in eliminating inconsistencies between business and stakeholders.
- [87] proposed a framework to prioritize requirements based on costs and benefits, by employing AHP. These author's approach is a multi-layered one and is designed for the domain of enterprise resource planning systems and their specific levels of requirements.
- [46] discussed goal-oriented requirements elicitation processes and explored how to select and prioritize the requirements using AHP by considering cost and effort criteria. A method, namely GOASREP, is proposed for the prioritization of the elicited requirements using an improved version of AHP.
- [108] evaluated seven requirement prioritization techniques and identified several limitations related to time complexity, lack of

scalability, inconsistency ratio, conflict among stakeholders. To overcomes these limitations, a prioritization model called ANN Fuzzy AHP is proposed. The model is evaluated on the case study of the supplier selection problem.
- [72] proposed a method, which is a combination of point direct scoring (PDS) and AHP and Kano model. After evaluation with the help of the Kano model, It is found that newly proposed approach namely ”T4” can help for obtaining a higher accuracy rate.

3.1.2. Requirements prioritization approaches based on fuzzy logic

To deal with the uncertainty of stakeholders' judgments, their expectations and software development process, 9 prioritization frameworks are found in literature. These frameworks are based on fuzzy logic introducing the concept of partial truth, where the truth values can only be an integer values 0 or 1.

- [71] suggested a framework that formulates the fuzzy definitions of software goals, desired situations and number requirements, which are derived from linguistic variables.
- [35] proposed a fuzzy logic-based approach in which the attributes of requirements are modelled as fuzzy variables. These fuzzy variables are integrated to perform prioritization decisions.
- [96,97][1] proposed two fuzzy logic-based RP methods to deal with the imprecise nature of requirements. However, unlike the other techniques presented in this section, the two techniques do not solely depend on fuzzy logic for prioritization. Instead, the multilevel prioritization is performed in which stakeholders and experts are equally involved. After the requirements are prioritized by stakeholders and experts, the requirements are fed into automated fuzzy logic system which is based on F-c-Mean (FCM) algorithm that prioritizes requirements clusters instead of single requirements.
- [2] identified inaccuracy as a main concern for the requirement prioritization and proposed an approach based on the use of fuzzy multi-criteria decision making in order to improve the accuracy. Each of the requirements is prioritized by its local weight and aggregated global weight in the decision matrix.
- [44] proposed an approach based on Yager's algorithm aimed at fusing the preference orderings by multiple stakeholders. This technique is applicable in case of individual preference orderings and hierarchical importance ranking. It is evaluated on realistic application example concerning the prioritization of requirements in the design of an aircraft seat.
- [102] reviewed extensive literature review and identified that the stakeholder identification method is not sufficient to prioritize the stakeholders need. A method is proposed for the prioritization of stakeholders on the basis of software requirements using fuzzy-based approach and evaluated with the help of a case study.
- [103] proposed the URPCalc method for RP of evolutionary requirements that leverages time series forecasting using fuzzy techniques. This approach is demonstrated to help product managers make decisions on the selection of requirements relevant for users in the future for the next product release.
- [7] proposed a fuzzy-based MoSCoW method for software requirements prioritization (SRP) and evaluated it with the help of Library Management System, as a case study. They have found that Fuzzy-MoSCoW method is efficient then AHP method.

3.1.3. Requirements prioritization approaches based on Stakeholders' preferences

From the recently published literature, we have found 8 studies that mainly focus on stakeholders for prioritization purposes. The list of stakeholders-preference-based RP approaches are as follows:

- [107] proposed a sample selection algorithm that helps in conducting a number of surveys with the customers. The algorithm suggests choosing the subset of requirements that can represent the whole pool of requirements. Each requirement is ranked by number of customers, where the final prioritized value is assigned by calculating arithmetic average mean and confidence intervals.
- Similarly, [77] proposed a RP approach that predicts user behaviour by maintaining the web server logs. This technique elicits requirements by analyzing the customers' usage behaviour for some specific application. The factor values like specific page hits, average time spent on a page, unique visits, etc, specify new requirements with their ranked value. The log-based data from number of customers is finally prioritized by average mean values.
- [38] used group recommendation heuristics for the RP purpose. They proposed a method in which each requirement is prioritized by many stakeholders. After the prioritization value is defined, the least distance, majority voting, average value and random selection heuristics are used to define the final rank. The decision phase calculates the precision of ranking by a mathematical formula.
- [105] suggest the use of social networks to actively involve the stakeholders in the process of requirement prioritization. They perform RE activities based on social network such as elicitation, prioritization and negotiation. The proposed approach was applied in three studies where participants used Facebook to participate in RE activities of different projects.
- [70] proposed a novel technique, the Majority Voting Goal Based (MVGB) technique for RP, which focused on the deadlock situation in RP. It discusses deadlock situation by satisfying the most appropriate stakeholders' requirements for achieving specific goals.
- [31] did an exploratory study to uncover how practitioners in a very large software project organization perform requirements (re) prioritization at inter-iteration time in the context of agile system delivery. These authors found that in large scale agile projects, the prioritization criteria are business value and risk. Moreover, the requirements prioritization process rests not only on user stories but also on other artefacts (i.e. delivery stories in this case, which provide information on the technical implementability of the requirements. In its exploration, the authors take the perspectives of the stakeholders' in the vendor's organization, however the requirements prioritization process strives to balance this perspective with the one of the client's organization. Because the paper reports exploratory research, it does not propose a specific method, but just a descriptive study on how requirements are prioritized in large scale agile and distributed projects.
- [82] proposed multi-criteria requirements prioritization for complex software applications and services, software evolution along with different stakeholders' perspective. Thus proposed a complex decision-making solution with automated reasoning techniques while considering user feedback.
- [106] set out to overcome dependencies between stakeholders requirements and proposed the DRank method in which an attributes tree is constructed to make stakeholders' ranking criteria selection easier and more operationalizable. PageRank, a basic mathematical algorithm to analyze the popularity of a webpage. [34] is proposed to analyze the dependencies between requirements. These authors used a simulation experiment to evaluate the DRank method by considering requirement dependencies.

3.1.4. Requirements prioritization approaches based on advanced data processing techniques

We observed that recent publications on RP methods exploit advanced data processing concepts such as machine learning and data repository-mining. Mostly, data is extracted for RE as done by data mining applications, but the help of machine learning, data is used for self-learning of software. Through machine learning, patterns in data are identified and applied accordingly. These approaches were deemed

[1] Counted them as two studies

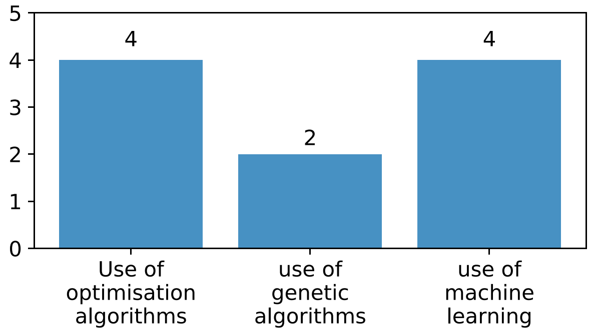

Fig. 4. Requirements prioritization approaches based on advanced data processing techniques.

very effective due to the automation and tooling supporting their implementation. We have found 10 studies that are based on advanced data processing techniques as can be seen from Fig. 4

1. RP approaches based on optimization algorithms.
   - [74] proposed a RP method that leverages meta-heuristic optimization algorithms, namely the Whale optimization and Grey wolf optimization algorithms. This method combines the advantages of each algorithm in order to prioritize the software requirements. The authors empirically evaluate their proposal's accuracy by using data from a large real-world project (of an access, library and ID card control system).
   - [11] designed a RP method based on the Whale optimization algorithm and demonstrated its superiority to the AHP.
   - [13] put forward the WhaleRank approach that rests on the Whale optimization algorithm. WhaleRank combines four ranking functions based on dictionary words, similarity measure, managers' perceptions, and the newly updated requirements, in order to yield a linear rank.
   - [49] proposed a collaborative requirement prioritization approach (called CDBR), which uses the Particle Swarm optimization algorithm in combination with linguistic values, and execute-before-after (EBA) relations among requirements. These authors' method help counter three major constraints in RP: dependencies among requirements, communication among stakeholders and developers and the issue of scalability.
2. RP approaches based on matrix algebra and genetic algorithms.
   - [78] introduced multi-dimensional matrices (the so-called tensors) to model the interactions between the functional requirements, the non-functional requirements and the stakeholders. The concepts of the matrix algebra (e.g. tensor decomposition, a multi-way analysis) are then employed to compute the final rank of each requirement.
   - [8] proposed a new technique to performs an enhancement in performance by using a least-squares-based random genetic algorithm for requirement prioritization. Main objective was to assist engineers in requirement prioritization with less time and less decision-making efforts.
3. RP approaches based on machine learning.
   - [90] presented one of the most stable and efficient approaches to record the stakeholders' preferences. A Case Based Rank (CBRank) method is proposed that provides the trade-off between elicitation effort and accuracy. Once the requirements are elicited, the rank function, target rank and final approximation rankare defined. Due to continuous machine learning process based on decisions makers' choices, these functions are able to perform much better after little iteration.
   - Similarly, another machine learning-based approach is proposed in [33], where requirements are automatically clustered based on predefined theme of business goals and stakeholders' interest. Non-functional Requirements (NFR) are also identified and prioritized by a tool that works on the principle of data mining called, NFR classifier. The partial automation of the prioritization process reduces the burden from software development team as well as improves reliability and efficiency.
   - [61] proposed three data-driven approaches i.e., individual attribute ranking, weighted ranking and regression ranking, to prioritize the feature improvement request for an application. Frequency, rating and emotion of user comments are considered prioritization features.
   - [3] proposed a data-driven requirement prioritization approach based on a weighted scale. The suggested approach combines a single decision matrix to provide the aggregated weight of each requirement. The implemented tool is evaluated to be scalable, efficient and highly reliable.

3.1.5. Requirements prioritization approaches based on numerical assignment

Based on our categorization, we have found 3 studies that exploit numerical assignment, which is grouping requirements into different priority groups as an associate step of requirement prioritization. The list of Numerical Assignment (NA) based RP approaches are as fellows.

- [75] proposed a technique that is specific for the prioritization of legal requirements in e-government projects. The implementation order of legal requirements comes from legal domain knowledge in order for software systems to comply with the laws and regulations. These requirements are mapped with legal text; based on number of mapping with sub-sections, number of cross-references and number of exceptions, the priority value is assigned. The lower score of requirement shows the readiness for implementations while higher score shows the need of further refinement.
- [73] proposed a method that uses the NA concept and adapts the factor analysis technique into requirement prioritization process in order to spot the hidden factors that influence the prioritization process. Factor analysis discovers several attributes from elicited requirements. The number of found attributes varies from requirements to requirements. Once the attributes are collected, analysis is performed to find out the associated attributed. A requirement attribute sheet is formalized to list down all the attributes relevant to associated requirements. Using the NA, the requirement attribute sheet is prioritized based on a numerical scale.
- Similarly, [114] proposed an extension to NA, which aims to reduce the imprecise nature of human judgment for prioritization process. To provide a space for imprecise judgments from stakeholders, the NA is , where extened NA receives the uncertain, imprecise and incomplete inputs in the form of group intervals and probability distributions. The aggregated input of extented NA is further refined by internal evident reasoning algorithm. When requirements are refined and tuned, the utility theory performs final ranking.

3.1.6. Requirements prioritization approaches based on B-Tree

From collected literature, we have found that only two methods employing the B-tree concept as its RP basis, where requirements are stored and retrieved in a balanced tree structure (See Fig. 3). These methods are continuing research of Beg et al. in year 2008 [19] and then 2009 [18] with a primary aim at achieving a reduction in number of comparisons for prioritization by relying on B-tree. In B-tree construction, the branching factor of requirements is considered which is reciprocal to number of comparisons. Standard B-tree algorithms are used where the prioritization is performed through search algorithms. However, further studies by Beg et al. [18]., proposed the use of a binary search algorithm to reduce number of comparisons because of the reduction in search space.

3.1.7. Requirements prioritization approaches based on cost-value approach

Cost-value approaches to prioritize the requirements are also used by 4 studies, described as follows.

- [68] proposed a cost value based technique in which requirements are first prioritized by stakeholders against their specific business goals. After the requirements are prioritized based on the benefit and cost approach of business, it is further prioritized by experts using the collaborative requirement management tool i.e., winBook. The TOPSIS decision analysis is used in winBook that assigns the final rank to specify the order of implementation.
- Similarly, [65] discussed resource efficiency in terms of requirements. They defined a cost function in order to prioritize the different requirement nodes based on their goals. The numerical results method was used for evaluation.
- [80] suggested the improvements in cost-value approach as well as in hierarchical cumulative voting approach. This approach supports the hierarchical structure for requirement prioritization as well as anticipates the cost, risk, impact and market driven aspects of requirement. The requirements are scaled from 1 to 10 weighted values against the business goals. The decomposition of requirements leads to the identification and final prioritization of its features.
- [9] proposed a cost-value technique that consider the dependencies among requirements within a requirements specification. In particular, the author accounts for dependencies among non-functional requirements.

3.1.8. Requirements prioritization approaches based on binary inputs

We found only one method based on binary inputs in literature. Otero et al. [86] proposed an evaluation framework for quality attributed to requirements. Due to intense conflicts in quality attributes, the binary inputs are used that only gauge the existence or absence of quality features in requirements. It is based on the identification of all quality attributes and features, where an order of implementation is defined by desirability function.

3.1.9. Requirements prioritization approaches based on mutation testing and dependability analysis

Our literature study indicated one method [26] that employed mutation testing analysis and dependability analysis to prioritize functional requirements. The method helps determine the importance of each requirement based on its criticality. It uses measurements of test suite adequacies and dependency values between faulted test cases and related requirements.

3.2. Evidence produced in evaluations of requirement prioritization techniques (**RQ2**)

One observation that we noticed from the selected literature (102 papers) is that RP researchers had provided an evaluation of their newly proposed approaches. The studies either compare one RP method with another, analyzing their limitations and those issues that affect the RP process, or provide a descriptive account of an application of the respective RP method and share some first lessons learned. Some empirical studies ([3,33,61,90]) do quantify the effectiveness of certain methods through an experiment and propose an improved framework. We have categorized the 102 publications in four classes taking four different perspectives: (1) comparative studies that empirically evaluate two or more RP methods and discuss which one is better in what context, (2) limitations-focused studies, (3) studies focused on contextual aspects of the projects where RP takes place, and (4) studies that explicitly deal with RP in agile software development. The next subsections present our findings in these four perspectives in more detail.

3.2.1. Comparative studies on requirement prioritization techniques

We have found 15 studies that compare the existing RP techniques based on their accuracy, ease of use, time consumption, degree of automation, subjective reliability and scalability. Table 1 presents the comparative studies with their respective criteria of comparison and results. Out of the 15 studies, 13 compared the AHP with at least one other RP method. Three out of those 13 studies dealt with the AHP and the Numeric Assignment based methods [29,60,62]. Regarding the criteria used to compare RP methods, we observe in Table 1 that ease of use (used in 9 out of the 10 studies) and scalability (used in 06 out of the 10 studies) are the most treated. [55] compared AHP with their own method ReDDCahp. [47] proposed an approach that uses Intuitionistic Fuzzy Sets considering the hesitation degree of decision-makers in the RP process. These sets represent the level of indeterminacy/hesitation due to unknown and indistinguishable ranks of requirements.

3.2.2. Studies focused on limitations of requirement prioritization techniques

We have found 6 studies that discuss the limitation of particular RP techniques in respective application domains. These studies explore the limitation of a particular technique by means of experiments or case studies and also suggest solutions to work around those limitations.

The limitations of the 100$ method and the binary search tree techniques are discussed with respect to globally distributed stakeholders in the study of Ahmad et al. [5]. These authors found that a change in the assigned value of requirement prioritization becomes complicated with 100$ and binary search tree technique in the project context where the stakeholders are globally distributed.

[6] also discuss the limitations of AHP with respect to globally distributed stakeholders. In this study, the authors conducted the experiment on a specific class of products that are usually used by clients worldwide e.g. mobile phones. The authors found that fulfilling the need of a large set of stakeholders exerts an immense pressure on the RP process. Both of these studies modified the 100$, the binary search tree and the AHP technique in order to consider the business aspects e.g., target market, affordability, competitive advantages during RP process.

Furthermore, [15] exposed the fact that most RP techniques perform efficiently and accurately when applied to a small number of requirements but fail to perform significantly when applied to large chunk of requirements. For example, the scalability issue is well known for the most prominent and accurate RP technique - the AHP, which becomes time-consuming and complex when requirements exceed the limit of 20. As AHP, the priority group and the binary search tree techniques are error-prone, they hence provide unreliable results in cases of large sets of requirements. However, in such contexts, Requirements triage, Requirement prioritization model, Fuzzy logic based RP are the processes that aim to perform efficiently. Moreover, [66] proposed a RP process that includes the sequential application of multiple well-known techniques (AHP, CV) in the context of a tendering process that deals with Requests For Proposals send by a client organization to software vendors. Using a real-world project, the author demonstrates the applicability of the process.

Similarly, [85] identifies the limitations of the AHP technique when applied on a large number of requirements. The scale-up limitations of AHP results are expressed in terms of increased complexity and reduced efficiency when the number of requirements are large. These authors present a model to mitigate this issues by combining the NA and the AHP where the NA categorizes the requirements into critical and optional, and the AHP is applied only on the particular critical set of requirements.

[1] also identifies the limitations of RP techniques and proposed a framework for combining RP approaches in a way that keeps the effect of their limitations to a minimum level. Their framework suggests that combining the 100$ method and Top 10 method would be useful in order to reduce the number of alternative requirements, while combining Win-win model and ranking is good to rank the aspects of requirements. The prioritized requirements with its relative aspects are

Table 1
Studies published after 2007, that compare requirement prioritization techniques.

Compared methods	Criteria of comparison	Result
AHP vs. NA [29]	Accuracy, Ease of use	Between AHP and NA, AHP is more accurate
AHP vs. CBRank [89]	Ease of use, Time consumption, Accuracy	CBRank in better regarding the first two aspects, but AHP is the best for being accurate
AHP vs. CV vs. LST [104]	Time duration, Scalability, Accuracy	LST (Likert Scale Technique) is the winner in efficiency while CV is more scalable. AHP is the winner in terms of accuracy and precision, while CV is ranked second.
BPL[a] vs. Wiegers' approach [17]	Time consumption, Ease of use, Reliability	BPL wins in all defined aspects but only when the number of requirements are small
Wiegers' approach vs. AHP vs. Fuzzy AHP vs. TOPSIS [113]	Complexity, Ease of use, Automation	Asymptotic analysis yields the Wiegers' approach and TOPSIS as the winner in terms of the least complex. Regarding ease of use, Wiegers' approach is the most formal approach while regarding automation, the AHP and fuzzy AHP are found to be the least automated.
NA vs. AHP vs. VOP vs. BST vs.PG [60]	Fault tolerance, Complexity, Time Consumption	AHP is the most fault tolerant but very complex while NA has the least fault tolerance but is very easy and quick
NA vs. AHP vs. ENA [115]	Avg. time, Ease of use, Attractiveness	ENA (Extensive numerical assignment) is better than NA and AHP, in terms of effectiveness, informativeness and efficiency
AHP vs. HAHP vs. Spanning tree vs. Bubble sort vs. Binary search vs. Priority group [62]	Ease of use, Reliability of results, Fault tolerance	AHP provide the most reliable results based on ratio scale
AHP vs. ELECTRE [41]	Ease of use, Scalability, Efficiency	ELECTRE I is easy to use and scalable. However, AHP is found to be more efficient
IPA vs. AHP vs. HAM based [28]	Accuracy of results,	Integrated prioritization approach (IPA) compared with AHP-based approach, and IPA was compared with the HAM-based approach.
MPPT vs. AHP vs Wieger's approach [56]	Efficiency, Understandability, Ease of use, Scalability, Time consumption	The MPPT is more effective, understandable, less time-consuming, more scalable and easier for prioritizing requirements than the AHP and and Wiegers' techniques.
ReDCCahp vs. AHP [55]	Efficiency, Understandability, Ease of use, Scalability, Time consumption, Effort consumption, Preference in practice	Between ReDCCahp and AHP, ReDCCahp is more scalable, with lower effort, shorter time, lower complexity, and higher understandability.
Tensor's method vs. AHP [78]	Accuracy, Ease of use, Time consumption	Tensor's method is superior to AHP regarding ease of use and time consumption, and as accurate as AHP.
CDBR vs. AHP vs IGA [49]	Accuracy, Effectiveness, Efficiency, Scalability, Processing time and Disagreement concerns among stakeholders and developers.	CDBR is superior to AHP and IGA in terms of efficiency and processing time. CDBR is comparable to AHP and IGA regarding the other criteria.
WhaleRank vs. CBRank vs Genetic Algorithm [13]	Accuracy, Disagreement measure	WhaleRank is superior to CBRank and GA.

[a] Binary Priority List

further prioritized by AHP. The combined methods reduced the number of requirements tremendously, thus resulted in reduced pair-wise comparisons by AHP.

3.2.3. Studies on contextual aspects of requirements prioritization

We found 9 empirical studies that analyze RP process from the perspective of the project context and treat contextual aspects such as the initial order in which the requirements are identified and presented to stakeholders, the type of information available during RP, the management of stakeholders' cognitive profiles and the use of consensus-driven decision-making as part of RP.

[109] performed an experiment to investigate the effect of initial order of requirements in RP process. The experiment concludes that the most important and least important initial order doesn't affect the RP process as the RP engineers are able to look beyond the initial ordering of requirements.

[10] proposed a framework called X2X specifically designed for startups. It is about creating a mutual communication ground area between business people, product managers/business analysts and technical team. Requirements are prioritize based on aspects such as

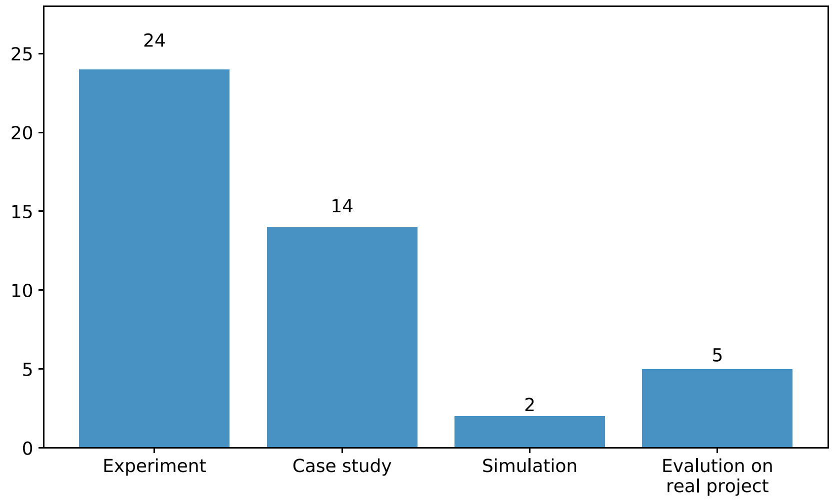

Fig. 5. Evaluation methods with respect to empirical studies.

core requirements, supplementary requirements, critical requirements.

Moreover, the experiment in [52] examines the type of information available at different stages of RP process. This study suggests the use of different RP methods at different stages of the RP process. If requirements are certain and reached an extent where it is unlikely to do further prioritization, then a simple ranking method is recommended, while if the requirements are highly volatile, the study recommends the use of AHP and 100$ methods.

[23] suggested to maintain the cognitive profiles of stakeholders, which is deemed to facilitate the understanding, representation and consensus process of RP for stakeholders. Contrary to the findings in [23], the study of Felfernig et al. [39] recommended to maintain the anonymous preference profiles of stakeholders in order to improve the quality of RP in terms of consensus, prioritization diversity and decision making process. A distributed tool is being used in the study of these authors for prioritization while keeping the involved stakeholders anonymous.

To understand the process of prioritizing quality requirements, Svensson et al. [110] conducted an exploratory case study on eleven companies. The study concludes that adhoc RP and priority groups are extensively used methods for quality requirements prioritization. Moreover, [35] performed a simulation study to analyze the effect of prioritization on traditional plan-based and agile based software development method. Furthermore, [98] and [54] focused on business aspects of requirement prioritization and risk based factors, respectively. Different models have been introduced and applied during prioritization discussed from more reliable basics. Focus was to reduce unnecessary (RE) activities and highlighting important requirements.

3.2.4. Studies on requirements prioritization in agile software development

During our evaluation of empirical studies' literature, we have also considered those studies that focused on prioritization and re-prioritization on requirements in agile projects. These studies did not put forward a particular prioritization approach, but attempted to gain a rich understanding of how agile requirements prioritization works in real-life projects and what candidates for 'good practices' could be suggested to the agile community. Racheva et. al. propose a descriptive model of concepts to consider when analyzing requirements prioritization in agile projects [91–93]. These authors' work takes the client's perspective on (re)prioritization in agile. The authors' conceptual model specifically considers the effort estimation, projects constraints, knowledge information and business value per feature from project manager's and client's perspective. The conceptual model is further evaluated for validity by means of a case study in 8 international companies which concludes that the practitioner's interpretation of business value per feature considerably varies from client to client [94]. Furthermore, the exploratory study of Daneva et al. [31] yielded a list of practices that large scale agile project organizations could possibly consider for balancing the vendor's and the client's perspective when prioritizing and re-prioritizing the requirements at inter-iteration time.

Next, [95] highlighted the issue starvation problem in agile software development, where due to lack of time and human resources, few issues always remain unattended until all the issues with higher priority have been accommodated. A new requirement prioritization approach based on priority score, aging score, and aging weight has been proposed to prevent issue starvation problem and to prioritize requirement efficiently and effectively.

Moreover, Yaseen et al. [119] suggest the iteration process model as part of the requirement prioritization process. A graph-based approach is adopted to explicate the dependencies among the requirements and sub-requirements. Finally, the spanning tree and numerical rating are used prioritize (dependent) requirements. The significance of the iteration model for requirement prioritization is evaluated by the requirements of mobile shop inventory management system. Sachdeva [101] proposed a novel multi-phase solution with the focus to maximize return-on-investment, while estimating the requirements priorities using Planning Poker, and modified Fibonacci series cards. The solution approach suggests maintaining separate product backlogs for customers facing requirements and technical debt of product owner. The method is evaluated on a software project that is used, within a multinational IT organization, to maintain test cases and their results.

3.3. Empirical evaluation approaches to validate RP methods (RQ3)

We found 45 papers that were dedicated to the validation of RP approaches. Fig. 5 shows the evaluation approaches that were used in the empirical studies that dealt with validation or evaluation of RP techniques. As we could see in Fig. 4, to validate RP approaches, the controlled experiment and case studies are the most prominent evaluations methods employed. However, we have noticed that there is ambiguity in meaning of case studies and experiments from many researchers' point of view. For the purpose of our research, we adopted the definitions of controlled experiment and case study as stated in [118]. Therein, 'experiment' is referred to as research-in-small which is under controlled setting, while 'case study' is referred to as research-in-typical which is based on monitoring real time projects, assignments, from industry/companies.

As shown in Fig. 5, most of the empirical studies conduct an experiment to evaluate their hypothesized claims while only one study

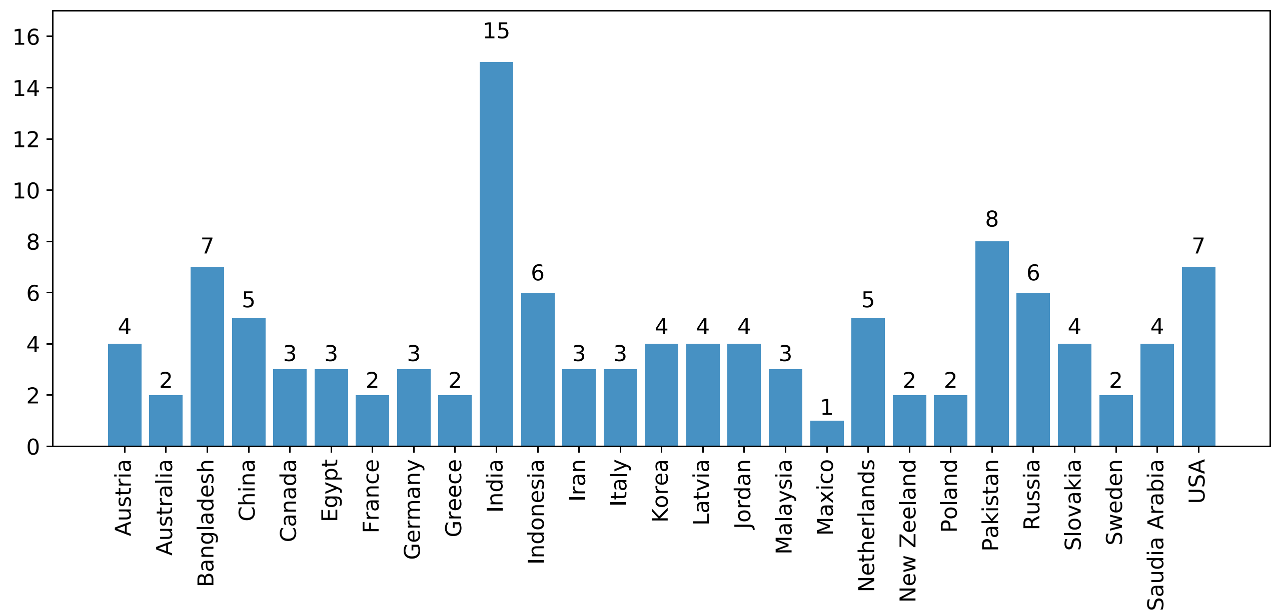

Fig. 6. Demographic trend of publication.

performed simulations. Out of 35 empirical studies (see Fig. 5), 20 studies performed experiments [5],[6],[23],[29],[37–39],[52],[54],[62],[69],[85],[89],[104],[109],[113],[115]. Twelve out of the 20 experiments are conducted in academic setting while six are carried out in real-life settings and with real stakeholders from industry. Some studies also utilize web-based tools (e.g. INTELLIREQ) in order to perform an experiment in distributed settings as well. We note that in experiments in academic settings, the students are considered as subjects. We have noticed that the sample size of students varies a lot: from 219 students to just 8 students. These experiments categorize the students participating in the experiment into class room and research students, in order to analyze their potential differences of preferences in the RP process. However, Carver et al. [24] warn about the use of students as subjects for research experiments due to the possible validity threats that the students' participation pose, e.g. students have varying background, work experience, and they differ in their willingness to participate. Furthermore, we have found one only experiment that used industrial stakeholders to compare the RP techniques [113].

There are some empirical studies conducted in industrial settings, too. These are of exploratory nature: they either compare the existing RP methods or propose new models based on discovered limitations [14,17,53,93,94,98,110,121]. The empirical studies that are specifically focused on agile development contexts are about real life data from RP process [7,72]. These series of case studies are used by similar group of authors, which further propose the conceptual model for RP in agile software development process. Simulation is also performed by two empirical studies. The first study measures the intangible and un-observable aspects of RP process e.g. value, importance, relative cost [35]. The second study [65] prioritized functionalities of smart grid nodes while keeping in view the cost function.

Finally, we observe a trend for conducting evaluation studies by using contexts and cases already described in literature. Five studies fall in this category [66],[103], [9],[74],[11]. For example, Krishnan [66] employs a case of the City of Duncan Website Redevelopment to illustrate the application of its proposed RP approach, while Alawneh uses a documented case of IBM [9].

3.4. Demographics of the studies per country

This subsection presents the countries in which the RP research was done in the set of our 102 selected papers. Fig. 6 reports on this. We observe that a nearly half of the RP publications were authored by researchers from Asia. For example, authors from India published 15 papers out of the 102 included in our SLR. Other examples are Pakistan (8), Bangladesh (7), and Indonesia (6). The variety of Asian countries indicates that the RP research is no longer limited to Western countries (West Europe and America), but has become more inclusive.

4. Discussion

Our review found that the AHP method is the most used one. Perhaps, this finding is unsurprising knowing that the AHP method is one of the oldest in the history of decision-making in software engineering and information systems. A recent bibliometric-based survey [122] on AHP techniques found that the research output with respect to AHP as a multiple criteria decision analysis technique outnumbers the research output concerning any other approach. In the area of RP, it seems that the AHP method was adopted by researchers as a 'benchmark' against which they to compare their newly proposed RP methods with.

We found a variety of criteria being used by researchers in comparing RP methods, plus heightened attention to the role of context in which the comparison takes place. Our findings (Table 1) allow us to think that there is a trend indicating that researchers are now much more focused on understanding which RP method works better in what context than this was the case 12 years ago. We think this indicates the growing maturity of the RP field.

An interesting observation of our review of the selected 102 studies is that there are no studies explicitly dedicated to the empirical evaluation of agile RP approaches. This finding is surprising, given the fact that the agile community of researchers is very active in investigating Requirements Engineering topics [57]. As the agile methods have a 15 years old history now, one might expect to find articles comparing agile RP techniques such as the Game and the Kanban in regard to criteria such as effectiveness and ease of use. We, however, did not find any empirical papers on such comparisons so far. A possible explanation for this observation could be that the agile philosophy is born in an environment in which practitioners' thinking takes the lead, with researchers following up, which in turn results in exploratory studies that help making practitioners' knowledge explicit and not experimental studies focused on 'what method works better in context'. Another reason could be that the agile community stands for simplification and light-weight approaches to software development, which implies that the RP techniques they deploy are also simple and light-weight and their ease of use is considered obvious. However, to know for sure what is the case we think that survey research would be necessary to shed light on the current RP practice in agile.

Next, in the 102 articles included in our review, we found that experiments were the most used empirical research method. This is in contrast to findings reported by other authors (e.g. [30]) suggesting that the most commonly used research method in the field of Requirements Engineering in general, is the case study. We think this difference is due to the fact that RP authors dealt with the empirical comparison between two or more methods much more frequently than, for example, authors of empirical papers in requirements elicitation or requirements modelling [32]. A related finding in our review is the fact that with very few exceptions the experiments in RP were carried out with student participants and not practitioners. This poses a question about the realism of the empirical evidence produced in these experiments. Therefore we think that more research is needed with practitioners experienced in RP to understands if a RP method that is found to be better than another method in a students' experimental setting would also turn out to be better in a real-life project context. To achieve this, students experiments could be followed up by case studies in real-world organizations.

Last, our results (see Fig. 3) suggest that the newly proposed RP approaches employ fuzzy logic, which is a well-known way to deal with uncertainty. In the case of RP, fuzzy logic is specifically leveraged to deal with the uncertainty of emerging requirements during the development process [71]. In contrast to research dated until 2007, where researchers were more focused on designing RP methods that can perform efficiently before the development of actual software begins, we observe a shift in RP research towards requirements evolution, namely how to prioritize ever-changing requirements. We think that this preoccupation of researchers with the treatment of uncertainty in RP could be traceable to the overall realization in the software engineering community that requirements are never completely fixed and could be subjected to changes, which is in stark contrast with the assumption about requirements completeness and 'big upfront requirements' mentality that dominated the RP methods from the 70ties and the 80ties. Indeed, approaches such as AHP assume that objective values exist and could feed the process to compare requirements pairwise. The heightened interest in fuzzy-logic-based proposals in the past 10 years might therefore suggest a shift in the thinking of the empirical research community. In fact, the increased number of RP proposals based on fuzzy logic might mean an acknowledgment that the assumption of requirements completeness the might not be realistic in all cases.

We also observe the rise of RP proposals that leverage advanced data-processing technologies such as machine learning and repository-mining [90]. Using such technologies reflects the current trend of big data use in software engineering. Observing the expansion of data techniques in virtually every sub-are of software engineering makes us think that in the future one might expect more publications on RP methods based on big data. For example, RE researchers (e.g. [48]) are now actively working on crowd-sourcing-based approaches to requirements elicitation in which they employ user feedback repositories (e.g. from app stores) or activity logs from internet-of-things systems, to come up with requirements for the next releases of software products. Prioritizing these requirements is recognized as an important activity, which makes us believe that more output of RP research could be expected in the future.

Moreover, we observe intense research in investigating stakeholders preference and their business goals as part of RP research. This matches the overall trend in software engineering for explicit user involvement and business value realization.

Finally, our analysis of the countries, in which research on RP methods was conducted, brought a surprising result. In contrast to previously published SLRs on empirical RE research (e.g. the 2018 SLR of Ambreen et al. [12]) that indicate the United States, Germany and Australia as the countries with the highest frequency of publishing RE research, we found that Asian countries, such as India, Bangladesh, Jordan, Saudi Arabia, Pakistan, Indonesia, China, Korea and Malaysia (among others) are very actively involved in RP research. We consider this finding as a signal for a positive development rendering RP research more inclusive. Including more countries, cultures and organizational and project settings in which proposed RP methods are applied, would ultimately lead to increased generalizability of our knowledge of RP methods. In particular, Section 3.1.4. indicates that the majority of papers proposing RP methods based on advanced data processing techniques, are authored by researchers in India, Jordan, and Malaysia. This interest in RP research might well be traceable to the recent large-scale programs sponsored by the governments of Asia countries (such as India) to leverage big data in critical sectors, e.g. IT, e-government and e-healthcare.

5. Limitations and threats to validity

The main threat to the validity of this research lies in the collection of relevant literature. In the RP sub-area of RE, researchers may use different terms and keywords to describe their scientific output. This inconsistent use of terminology is a serious threat that might lead to the collection of inconsistent data. We mitigated this threat by defining the proper search strings in which alternative keywords, in addition to Boolean conjunctions, are used. The search strings are based on our research question and are iteratively executed, exploiting the advanced search functions of the digital libraries, in order to assure the strings' effectiveness. Furthermore, we also consulted the reference lists of the recently published studies in order to identify additional relevant publications. However, as already indicated, this snowballing step did not yield new publications to our set.

Another threat to the validity includes the authors' bias in the selection of studies and in the extraction of data from these studies. In our SLR, such a bias is possible due to the fact that one of the authors (Daneva) published empirical studies in RP. To counter this threat, we have formulated a set of quality evaluation questions (see Section 2.3) for the critical appraisal of each study (See Appendix A). We checked each study against the quality assessment criteria. We have selected only those studies that scored a minimum of 2 points on these criteria, which means the study answers 'yes' to at least 2 quality evaluation questions, out of our 4.

6. Related work

There are four literature studies [4,53,88,99] that form the related work and also the basis for our paper. Each of them has a specific goal, which is different from ours and, in tur,n has different research questions that examine RP research. The systematic review of Riegel and Doerr [99] examined the prioritization criteria used in published research on RP techniques and came up with a prioritization criteria model. While the topic of these authors review relates to ours, its research goal and ours differ in that we attempt to identify trends in RP proposals in general and these proposals' applicability in context from 2007 onward. As indicated earlier, our research draws on the review of Herrmann and Daneva [53]. We extended the observations from this study (which covers publications until 2007), by identifying trends in empirical RP research in the period 2007–2019. However, in contrast to [53] which t focused on the cost and benefit related criteria of RP methods, our focus in the present SLR is on the evidence produced in empirical studies that suggest those contexts in which a RP method is applicable to.

Next, the study of Achimugu et al. [4] focused on the prioritization scales and those limitations of the existing RP techniques that are inherent to the design of each RP method. Our results extend the results of these authors, by pinpointing to aspects of the applications of RP techniques for which there is published evidence that the RP techniques actually worked.

Furthermore, the mapping study of Pergher and Rossi [88] took an empirical evaluation perspective on RP research and examined those evaluation criteria that have been used in the evaluation studies of RP methods. These authors found that accuracy, time consumption and ease of use were the top-3 criteria mentioned in the PR literature to evaluate a method. Unlike this mapping study, our research is focused on the objects of comparison and the contexts in which certain RP methods are more suitable than others. Specifically, we looked into what RP method is compared with what another method; we complement the results of Pergher and Rossi [88] by focusing on the 'how side' of the studies and looked at how exactly these evaluation aspects were studied. For example, while our observations agree with these authors regarding the use of accuracy as the most evaluated aspect of RP methods, we also found that accuracy is actually used primarily in studies evaluating the AHP method (see Table 1).

7. Implications for practitioners and for researchers

Our results indicate that when empirical researchers compare RP methods (see Table 1), the most frequently used criteria are ease of use and accuracy. We found these criteria in 8 out of the 10 method comparing studies in Table 1. Both criteria are highly relevant to practitioners. If practitioners are interested in extending their RP toolbox, Table 1 would inform them on what possible methods to consider for trying out in their next project. An interesting observation in Table 1 is regarding the AHP method. When compared to other methods on the criterion 'ease of use', AHP ranks second to the other methods being compared to (e.g. the Numeric Assignment method, CBRank or Wieger's method). However, when compared on the criterion 'accuracy', Table 1 indicates that the AHP method is superior to the other methods being compared to. To practitioners, this means that when accuracy is their most important criterion for choosing a RP method, they should consider the AHP as their first choice. However, if ease of use is what they look for, then other RP methods are better candidates to choose from.

Our paper has the following implications for research. First, we have found that the use of Fuzzy Logic for requirement prioritization purposes mitigates the issue of stakeholders uncertainty but at the same time leads to a tremendous increase in complexity of its applicability. We think that some automation would be beneficial in order to cope with complexity. This would be helpful for increasing the chance of a fuzzy-logic-based RP method to be used in practice. Therefore we suggest researchers focus on the automation aspects of their proposed approaches and the possible integration of the automated RP in RE tools.

Second, among the number of proposed methods and techniques, we observed a lack of definitions of the projects to which these approaches are most suited. In empirical software engineering, researchers categorize software projects into small, medium and large (e.g. [31]). However, with very few exceptions the majority of the RP publications in our SLR do not provide a definition of what they call 'large', nor they describe how 'large' is characterized. We think that researchers could increase the rigor in their empirical work by giving much more elaboration in their definitions of context. Furthermore, we acknowledge that this categorization (of small, medium and large projects) may not always be sufficient for the purpose of empirical studies in RP. Rich contextual descriptions of the projects beyond size aspects, are needed to understand why one RP method is, for example, easier to adopt or more attractive than another. More research in this respect is needed to gain more knowledge on the suitability of a RP method to particular projects, (see Tables 2 and 3 in Appendix A and Appendix B, respectively).

Last, we also observed a trend of adapting psychological decision-making tools in order to make the RP process efficient enough. Studies suggest that maintaining the cognitive profile of stakeholders can increase the effectiveness and quality of RP processes. It is easier for requirements specialists and business analysts to present the requirements in a form that is most acceptable and understandable for stakeholders in situations where the background knowledge of the stakeholders may vary tremendously. We consider the research on borrowing theories and approaches from cognitive science and social psychology for the purpose of RP method development an interesting line for future research.

8. Conclusions

This paper provides a structured understanding of the RP methods and their empirical evaluation that have been published between 2007 and 2019. We carried out a research process employing systematic literature review techniques, which resulted in the identification and analysis of newly proposed RP methods and their comparisons in terms of specific criteria, e.g. ease of use, effectiveness, scalability. Our most important observations are summarized and structured according to our RQs as follows:

***RQ1**: What are the published software RP methods/techniques/approaches in last decade (2007 to 2019)?* We found a total of 54 proposals for RP methods. Many of them were improvements of existing methods: the Analytical Hierarchical Process, the Numerical Assignment, the Binary Search Tree, the Planning Game, the Cumulative Voting, the Quality Functional Deployment method, the cost-value approach, the Win-Win method, MosCow, the Top10, the bubble sort approach and the Wiegers' approach. However, we found approaches that use machine learning and mining techniques or software repositories in order to prioritize requirements. It is our understanding that these RP methods reflect the current trend in the broad software engineering field to leverage the presence of big data in regard to software processes and artifacts. One might expect to have more RP research focused on machine learning and repository-mining in the future.

***RQ2**: What empirical evidence has been produced in scientific literature about RP methods/ techniques/approaches that were published after 2007?* We found 15 empirical studies reporting comparative evaluation of two or more RP methods. Based on our selected sources, we have found that AHP and NA are reported as the most accurate approaches. We make the observation that researchers used a variety of evaluation criteria (more than 10, see Table 1): accuracy, ease of use, scalability, efficiency, attractiveness, time consumption, fault tolerance reliability, complexity, level of automation. Despite the massive evaluation work, we could not find any study that clarifies what the best suited method is for a particular type of project. We therefore think that more research is needed to address this match between methods and contextual settings of projects in which they could be most useful.

***RQ3**: What evaluation approaches have been used in empirical studies to validate the proposed RP methods/approaches/techniques?* We found that experiments were by far the most used research method for empirical evaluations in RP. This came as a surprise knowing that case studies have been indicated as the most popular research method in the field of RE (e.g. see Daneva et al.[30]). With very few exceptions, the experiments reported in the literature sources in our SLR, are with students. Given the possible external validity threats that accompany experiments with students, we think that more research is needed to evaluate and compare RP methods in real-world contexts. Only then can we make generalizability claims about the applicability of a RP method.

Appendix A

Table 2
Quality assessment according to seven quality criteria.

Reference	QC1	QC2	QC3	QC4	QC5	QC6	QC7	Total score
[8]	1	1	0	1	1	1	0	5
[7]	1	1	0	0	1	1	0	4
[11]	1	0	0	1	1	0	0	3
[9]	1	1	0	0	0	0	0	2
[13]	1	0	1	1	1	0	1	6
[10]	1	0	0	0	1	1	0	3
[1]	1	1	0	1	0	1	0	4
[4]	0	0	1	0	0	1	1	3
[2]	1	1	1	0	0	1	0	4
[3]	1	1	1	0	0	1	0	4
[14]	1	1	0	1	0	0	0	3
[6]	1	1	0	1	1	0	1	5
[20]	1	1	0	1	1	0	1	5
[21]	1	1	0	0	1	0	1	4
[15]	0	0	1	0	0	1	1	3
[17]	0	0	1	0	0	1	1	3
[18]	1	1	0	0	0	0	0	2
[19]	1	1	0	1	0	1	0	4
[25]	1	1	0	1	0	0	0	3
[26]	1	1	0	0	0	0	0	2
[23]	1	1	0	1	0	0	0	3
[24]	1	1	0	1	0	1	0	4
[22]	1	1	0	1	0	1	0	4
[27]	1	1	0	1	0	0	1	4
[28]	1	1	1	1	0	1	1	6
[29]	0	0	0	0	1	1	1	3
[31]	1	1	0	0	1	1	1	5
[33]	1	1	1	1	0	1	1	6
[35]	1	1	0	0	1	1	0	4
[36]	1	1	1	0	1	1	0	5
[37]	1	1	1	0	1	0	1	5
[38]	1	1	1	0	1	0	0	4
[39]	0	0	0	1	1	1	0	3
[40]	0	0	0	1	1	0	1	3
[41]	1	1	1	1	1	0	0	5
[42]	1	1	0	1	1	0	0	4
[43]	0	0	0	1	1	1	1	4
[45]	0	0	0	1	1	1	0	3
[46]	1	1	1	1	1	0	0	5
[47]	1	1	1	0	0	0	0	3
[49]	1	0	1	1	1	1	1	6
[50]	1	1	1	0	1	1	0	5
[52]	0	0	0	1	1	1	0	3
[53]	0	0	0	1	1	1	1	4
[54]	1	1	0	1	1	1	0	5
[55]	0	0	1	1	1	1	1	5
[56]	1	0	1	1	1	1	1	6
[57]	0	0	0	1	1	1	1	4
[58]	1	1	1	1	0	0	0	4
[59]	1	1	1	1	1	1	0	6
[60]	0	0	0	0	0	1	1	2
[61]	1	1	1	0	0	1	1	5
[62]	0	0	0	1	1	1	1	4
[63]	1	1	1	0	0	0	1	4
[65]	1	1	0	1	1	1	0	5
[66]	1	1	1	1	0	0	0	4
[68]	1	1	1	1	1	0	0	5
[70]	1	1	0	1	1	1	0	5
[71]	1	1	1	1	1	1	0	6
[78]	1	0	1	1	1	1	0	5
[72]	1	1	0	1	0	1	1	5
[83]	1	1	0	1	0	1	0	4
[76]	1	0	1	1	1	0	1	5
[74]	1	1	1	1	1	0	1	6
[73]	0	0	1	1	1	0	1	4
[75]	1	0	0	0	1	0	1	3
[77]	1	1	0	1	1	0	0	4

(*continued on next page*)

Table 2 (*continued*)

Reference	QC1	QC2	QC3	QC4	QC5	QC6	QC7	Total score
[79]	1	1	1	0	0	0	1	4
[80]	1	1	1	1	1	0	1	6
[82]	0	0	1	1	1	1	1	5
[84]	1	1	0	1	1	1	0	5
[85]	0	0	0	1	1	0	1	4
[86]	1	1	1	1	0	0	1	5
[87]	1	1	1	1	0	0	1	5
[88]	0	0	0	1	1	1	1	4
[89]	0	0	1	1	1	1	0	4
[90]	1	1	1	1	0	0	1	5
[95]	1	1	1	0	1	0	0	4
[91]	1	1	1	0	0	0	1	4
[92]	0	0	1	1	0	1	1	4
[93]	1	1	1	1	0	0	1	5
[94]	0	1	1	1	0	0	1	4
[96]	1	0	1	1	0	0	1	4
[97]	0	1	1	1	0	0	1	4
[98]	0	0	1	1	1	1	1	5
[99]	0	0	0	1	1	1	1	4
[101]	0	1	1	1	1	0	1	5
[102]	1	1	1	1	0	0	1	5
[103]	1	0	1	0	0	0	0	2
[104]	1	0	1	1	1	0	1	5
[105]	1	1	1	1	0	0	1	5
[106]	1	0	1	1	0	1	1	5
[107]	1	1	1	0	0	0	0	3
[108]	1	1	1	1	0	0	1	5
[110]	0	0	1	1	1	1	1	5
[109]	1	1	1	1	0	0	1	5
[111]	1	1	1	1	0	0	0	4
[112]	0	1	1	1	0	0	1	4
[114]	1	0	1	1	1	1	1	6
[115]	0	0	1	1	1	1	1	5
[113]	1	0	1	1	1	0	1	5
[119]	0	1	1	0	1	1	0	4
[121]	0	1	1	0	1	0	1	4
[122]	0	1	1	1	1	0	1	5
[120]	1	1	10	1	1	0	1	5

Appendix B

Table 3
Examples of studies included in our analysis to answer RQ3.

Reference	**Validation method**	**Validation setting**	**Purpose of study**
[40]	Used a tool INTELLIREQ	Experiment 219 students	Analyse the factors that influence the RP process. major focus on preference decisions by group of stakeholders Anonymized RP improve quality of RP in term of prioritization diversity, quality of resulting s/w and team consensus.
[85]	Experiment	Tried to do with industry but done with 8 masters students	Limitations of AHP when applied on large no. of requirements, compare AHP and NA, propose a model that is based on AHP and NA that will reduce complexity and time consumption
[85]	Experiment	Students[Used software INTELLREQ]	Validate anonymous preference elicitation for RP by empirical study, it improves the quality of RP process
[113]	Experiment by fuzzy logic/case study	Industry stakeholders	Compare different RP methods on varying attributes, Analyse which RP method is best to meet the need of software. Employ fuzzy decision making process
[17]	Case studies	Two Dutch companies	Compare Binary P technique with Weiger's technique with two case studies. In term of ease of use, subjective reliability and time consumption, BPL can be used in small no. of requirements in small s/w companies
[93]	Case study based on questionnaire and interviews	11 interviews, 8 companies	Based on agile RP, Identify conceptual categories, mapping between categories and existing techniques e.g. aspects, value of prioritization, learning, etc
[110]	Case study with interviews with experts	11 companies	Focus on quality RP. State that only quality R are have by default low prioritization, analyse how QRP occur in practise.
[100]	Case study	Industry Data [propose model]	Prioritization by stakeholders, state of the art of existing RP techniques, find short coming and den propose reducing the unnecessary activities by focusing on P based on business objectives. Define that elicitation process need to be guided in order to have efficient RP
[15]	NONE	None[Present the extension to model]	Limitation of existing RP techniques, need of automation where requirements are abundant in numbers, provide fuzzy logic based solution for automation, extension in value based intelligent RP.
[60]	NONE	NONE [just compare]	Compare 7 RP methods based on complexity, scale, fault tolerance, time consumption etc

References

[1] M. Aasem, M. Ramzan, A. Jaffar, Analysis and optimization of software requirements prioritization atechniques, International Conference on Information and Emerging Technologies (ICIET), 2010, IEEE, 2010, pp. 1–6.
[2] P. Achimugu, A. Selamat, R. Ibrahim, Using the fuzzy multi-criteria decision making approach for software requirement prioritization, Jurnal Teknologi 77 (13) (2015).
[3] P. Achimugu, A. Selamat, R. Ibrahim, Reprotizer: a fully implemented software requirements prioritization tool, Transactions on Computational Collective Intelligence XXII, Springer, 2016, pp. 80–105.
[4] P. Achimugu, A. Selamat, R. Ibrahim, M.N. Mahrin, A systematic literature review of software requirements prioritization research, Inf. Softw. Technol. 56 (6) (2014) 568–585.
[5] A. Ahmad, M. Goransson, A. Shahzad, Limitations of the analytic hierarchy process technique with respect to geographically distributed stakeholders, World Acad. Sci. Eng. Technol. 69 (2010) 111–116.
[6] A. Ahmad, A. Shahzad, V.K. Padmanabhuni, A. Mansoor, S. Joseph, Z. Arshad, Requirements prioritization with respect to geographically distributed stakeholders, International Conference on Computer Science and Automation Engineering (CSAE), 2011, 4 IEEE, 2011, pp. 290–294.
[7] K.S. Ahmad, N. Ahmad, H. Tahir, S. Khan, Fuzzymoscow: a fuzzy based MoSCoW method for the prioritization of software requirements, 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), (2017), pp. 433–437.
[8] H. Ahuja, U. Batra, et al., Performance Enhancement in Requirement Prioritization by Using Least-squares-based Random Genetic Algorithm, Innovations in Computational Intelligence, Springer, 2018, pp. 251–263.
[9] L. Alawneh, Requirements prioritization using hierarchical dependencies, Information Technology-New Generations, Springer, 2018, pp. 459–464.
[10] S. Albuga, Y. Odeh, Towards prioritizing software business requirements in startups, 2018 8th International Conference on Computer Science and Information Technology (CSIT), IEEE, 2018, pp. 257–265.
[11] A. Alzaqebah, R. Masadeh, A. Hudaib, Whale optimization algorithm for requirements prioritization, 2018 9th International Conference on Information and Communication Systems (ICICS), IEEE, 2018, pp. 84–89.
[12] T. Ambreen, N. Ikram, M. Usman, M. Niazi, Empirical research in requirements engineering: trends and opportunities, Requir. Eng. 23 (1) (2018) 63–95.
[13] R.V. Anand, M. Dinakaran, Whalerank: an optimisation based ranking approach for software requirements prioritisation, Int. J. Environ. Waste Manag. 21 (1) (2018) 1–21.
[14] S.A. Asif, Z. Masud, R. Easmin, A.U. Gias, Saffron: a semi-automated framework for software requirements prioritization, arXiv preprint arXiv:1801.00354(2017).
[15] M.I. Babar, M. Ramzan, S.A. Ghayyur, Challenges and future trends in software requirements prioritization, International Conference on Computer Networks and Information Technology (ICCNIT), IEEE, 2011, pp. 319–324.
[16] E. Bagheri, M. Asadi, D. Gasevic, S. Soltani, Stratified analytic hierarchy process: Prioritization and selection of software features, International Conference on Software Product Lines, Springer, 2010, pp. 300–315.
[17] T. Bebensee, I. van de Weerd, S. Brinkkemper, Binary priority list for prioritizing software requirements, International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2010, pp. 67–78.
[18] M.R. Beg, R.P. Verma, A. Joshi, Reduction in number of comparisons for requirement prioritization using b-tree, International Advance Computing Conference, IACC, IEEE, 2009, pp. 340–344.
[19] R. Beg, Q. Abbas, R.P. Verma, An approach for requirement prioritization using b-tree, 2008 First International Conference on Emerging Trends in Engineering and Technology, IEEE, 2008, pp. 1216–1221.
[20] P. Berander, A. Andrews, Requirements prioritization, Engineering and managing software requirements, Springer, 2005, pp. 69–94.
[21] P. Berander, K.A. Khan, L. Lehtola, Towards a research framework on requirements prioritization, SERPS 6 (2006) 18–19.
[22] E. Carmel, R. Agarwal, Tactical approaches for alleviating distance in global software development, IEEE Softw. 18 (2) (2001) 22–29.
[23] N.M. Carod, A. Cechich, Cognitive-driven requirements prioritization: a case study, 9th IEEE International Conference on Cognitive Informatics (ICCI), IEEE, 2010, pp. 75–82.
[24] J. Carver, L. Jaccheri, S. Morasca, F. Shull, Issues in using students in empirical studies in software engineering education, Software Metrics Symposium, 2003. Proceedings. Ninth International, IEEE, 2003, pp. 239–249.
[25] K.Y. Chan, C. Kwong, T.S. Dillon, Determination of Importance of Customer Requirements Using the Fuzzy AHP Method, Computational Intelligence Techniques for New Product Design, Springer, 2012, pp. 59–77.
[26] N. Condori-Fernandez, M.F. Granda, T.E. Vos, Towards a functional requirements prioritization with early mutation testing, Proceedings of the 5th International Workshop on Requirements Engineering and Testing, ACM, 2018, pp. 21–24.
[27] M. Dabbagh, S.P. Lee, A consistent approach for prioritizing system quality attributes, 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, 2013, pp. 317–322.
[28] M. Dabbagh, S.P. Lee, R.M. Parizi, Functional and non-functional requirements prioritization: empirical evaluation of ipa, ahp-based, and ham-based approaches, Soft Comput. 20 (11) (2016) 4497–4520.
[29] A.S. Danesh, R. Ahmad, Study of prioritization techniques using students as subjects, International Conference onInformation Management and Engineering, 2009. ICIME'09, IEEE, 2009, pp. 390–394.
[30] M. Daneva, D. Damian, A. Marchetto, O. Pastor, Empirical research methodologies and studies in requirements engineering: how far did we come? J. Syst. Softw. 95 (2014) 1–9.
[31] M. Daneva, E. Van Der Veen, C. Amrit, S. Ghaisas, K. Sikkel, R. Kumar, N. Ajmeri, U. Ramteerthkar, R. Wieringa, Agile requirements prioritization in large-scale outsourced system projects: an empirical study, J. Syst. Softw. 86 (5) (2013) 1333–1353.
[32] M. Daneva, R. Wieringa, Requirements engineering for enterprise systems: what we know and what we don't know? Intentional perspectives on information systems engineering, Springer, 2010, pp. 115–136.
[33] C. Duan, P. Laurent, J. Cleland-Huang, C. Kwiatkowski, Towards automated requirements prioritization and triage, Requir. Eng. 14 (2) (2009) 73–89.
[34] H. Dubey, B. Roy, An improved page rank algorithm based on optimized normalization technique, Int. J. Comput. Sci. Inf. Technol. 2 (5) (2011) 2183–2188.
[35] A. Ejnioui, C. Otero, L. Otero, A simulation-based fuzzy multi-attribute decision making for prioritizing software requirements, Proceedings of the 1st Annual conference on Research in information technology, ACM, 2012, pp. 37–42.
[36] M.A.A. Elsood, H.A. Hefny, E.S. Nasr, A goal-based technique for requirements prioritization, Informatics and Systems (INFOS), 2014 9th International Conference on, IEEE, 2014, pp. SW–18.
[37] M. Fargnoli, N. Haber, A practical anp-qfd methodology for dealing with requirements' inner dependency in pss development, Comput. Ind. Eng. 127 (2019) 536–548.
[38] A. Felfernig, G. Ninaus, Group recommendation algorithms for requirements prioritization, Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering, IEEE Press, 2012, pp. 59–62.
[39] A. Felfernig, G. Ninaus, F. Reinfrank, Eliciting stakeholder preferences for requirements prioritization. Decisions@ RecSys, (2012), pp. 27–31.
[40] F. Fellir, K. Nafil, R. Touahni, System requirements prioritization based on ahp, Third IEEE International Colloquium in Information Science and Technology (CIST), IEEE, 2014, pp. 163–167.
[41] J.M. Fernandes, S.P. Rodrigues, L.A. Costa, Comparing ahp and electre i for prioritizing software requirements, 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, 2015, pp. 1–8.
[42] P. Fitsilis, V. Gerogiannis, L. Anthopoulos, I.K. Savvas, Supporting the requirements prioritization process using social network analysis techniques, 2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, IEEE, 2010, pp. 110–115.
[43] F. Franceschini, D. Maisano, Prioritization of qfd customer requirements based on the law of comparative judgments, Qual. Eng. 27 (4) (2015) 437–449.
[44] F. Franceschini, D. Maisano, L. Mastrogiacomo, Customer requirement prioritization on qfd: a new proposal based on the generalized yager's algorithm, Res. Eng. Des. 26 (2) (2015) 171–187.
[45] N. Garg, P. Agarwal, S. Khan, Recent advancements in requirement elicitation and prioritization techniques, International Conference on Advances inComputer Engineering and Applications (ICACEA), IEEE, 2015, pp. 237–240.
[46] N. Garg, M. Sadiq, P. Agarwal, Goasrep: Goal oriented approach for software requirements elicitation and prioritization using analytic hierarchy process, Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Springer, 2017, pp. 281–287.
[47] V.C. Gerogiannis, P. Fitsilis, G. Kakarontzas, C. Born, Handling vagueness and subjectivity in requirements prioritization, Proceedings of the 22nd Pan-Hellenic Conference on Informatics, ACM, 2018, pp. 150–155.
[48] E.C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, et al., The crowd in requirements engineering: the landscape and challenges, IEEE Softw. 34 (2) (2017) 44–52.
[49] A. Gupta, C. Gupta, Cdbr: a semi-automated collaborative execute-before-after dependency-based requirement prioritization approach, J. King Saud Univ.-Comput.Inf. Sci. (2018).
[50] V. Gupta, D. Chauhan, K. Dutta, C. Gupta, Requirement reprioritization: a multilayered dynamic approach, Int. J. Softw. Eng. Appl. 7 (5) (2013) 55–64.
[51] A.-W. Harzing, S. Alakangas, Google scholar, scopus and the web of science: a longitudinal and cross-disciplinary comparison, Scientometrics 106 (2) (2016) 787–804.
[52] S. Hatton, Choosing the right prioritisation method, 19th Australian Conference on Software Engineering (aswec 2008), IEEE, 2008, pp. 517–526.
[53] A. Herrmann, M. Daneva, Requirements prioritization based on benefit and cost prediction: an agenda for future research, International Conference on Requirements Engineering, RE'08, IEEE, 2008, pp. 125–134.
[54] A. Herrmann, B. Paech, Practical challenges of requirements prioritization based on risk estimation, Empir. Softw. Eng. 14 (6) (2009) 644–684.
[55] I. Ibriwesh, S.-B. Ho, I. Chai, Overcoming scalability issues in analytic hierarchy process with redccahp: an empirical investigation, Arab. J. Sci. Eng. 43 (12) (2018) 7995–8011.
[56] I. Ibriwesh, S.-B. Ho, I. Chai, C.-H. Tan, Prioritizing solution-oriented software requirements using the multiple perspective prioritization technique algorithm: an empirical investigation, Concurr. Eng. 27 (1) (2019) 68–79.
[57] I. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Comput. Hum. Behav. 51 (2015) 915–929.
[58] M.A. Iqbal, A.M. Zaidi, S. Murtaza, A new requirement prioritization model for market driven products using analytical hierarchical process, International

Conference on Data Storage and Data Engineering (DSDE), IEEE, 2010, pp. 142–149.
[59] M. Kassab, N. Kilicay-Ergin, Applying analytical hierarchy process to system quality requirements prioritization, Innov. Syst. Softw. Eng. 11 (4) (2015) 303–312.
[60] G. Kaur, S. Bawa, A survey of requirement prioritization methods, International Journal of Engineering Research and Technology, 2 ESRSA Publications, 2013.
[61] S. Keertipati, B.T.R. Savarimuthu, S.A. Licorish, Approaches for prioritizing feature improvements extracted from app reviews, Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, ACM, 2016, p. 33.
[62] J.A. Khan, I.U. Rehman, Y.H. Khan, I.J. Khan, S. Rashid, Comparison of requirement prioritization techniques to find best prioritization technique, Int. J. Mod. Educ.Comput. Sci. 7 (11) (2015) 53.
[63] S.U.R. Khan, S.P. Lee, M. Dabbagh, M. Tahir, M. Khan, M. Arif, Repizer: a framework for prioritization of software requirements, Front. Inf. Technol. Electron.Eng. 17 (2016) 750–765.
[64] B.A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based software engineering and systematic reviews, 4 CRC Press, 2015.
[65] V. Kouhdaragh, D. Tarchi, A. Vanelli-Coralli, G.E. Corazza, A Cost Function Based Prioritization Method for Smart Grid Communication Network, Smart Grid Inspired Future Technologies, Springer, 2017, pp. 16–24.
[66] M.S. Krishnan, Rfp based requirement prioritization–a one-step solution, Mater. Today 5 (1) (2018) 642–649.
[67] M. Kuhrmann, D.M. Fernández, M. Daneva, On the pragmatic design of literature studies in software engineering: an experience-based guideline, Empir. Softw. Eng. 22 (6) (2017) 2852–2891.
[68] N. Kukreja, Decision theoretic requirements prioritization: a two-step approach for sliding towards value realization, Proceedings of the 2013 International Conference on Software Engineering, IEEE Press, 2013, pp. 1465–1467.
[69] Y. Li, M. Zhang, T. Yue, S. Ali, L. Zhang, Search-based uncertainty-wise requirements prioritization, 2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS), (2017).
[70] R.M. Liaqat, M.A. Ahmed, F. Azam, B. Mehboob, A majority voting goal based technique for requirement prioritization, Automation and Computing (ICAC), 2016 22nd International Conference on, IEEE, 2016, pp. 435–439.
[71] D.C. Lima, F. Freitas, G. Campos, J. Souza, A fuzzy approach to requirements prioritization, International Symposium on Search Based Software Engineering, Springer, 2011, pp. 64–69.
[72] P. Madzík, L. Lysá, P. Budaj, Determining the importance of customer requirements in qfd–a new approach based on kano model and its comparison with other methods, Calitatea 20 (168) (2019) 3–15.
[73] S.A. Marjaie, V. Kulkarni, Recognition of hidden factors in requirements prioritization using factor analysis, International Conference on Computational Intelligence and Software Engineering (CiSE), IEEE, 2010, pp. 1–5.
[74] R. Masadeh, A. Hudaib, A. Alzaqebah, Wgw: a hybrid approach based on whale and grey wolf optimization algorithms for requirements prioritization, Adv. Syst. Sci. Appl. 18 (2) (2018) 63–83.
[75] A.K. Massey, P.N. Otto, A.I. Antón, Prioritizing legal requirements, Requirements Engineering and Law (RELAW), 2009 Second International Workshop on, IEEE, 2009, pp. 27–32.
[76] J. McZara, S. Sarkani, T. Holzer, T. Eveleigh, Software requirements prioritization and selection using linguistic tools and constraint solvers'a controlled experiment, Empir. Softw. Eng. 20 (6) (2015) 1721–1761.
[77] O.M. Mendizabal, M. Spier, R. Saad, Log-based approach for performance requirements elicitation and prioritization, 2012 20th IEEE International Conference on Requirements Engineering Conference (RE'12), IEEE, 2012, pp. 297–302.
[78] N. Misaghian, H. Motameni, An approach for requirements prioritization based on tensor decomposition, Requir. Eng. 23 (2) (2018) 169–188.
[79] J. Misra, S. Sengupta, S. Das, Latent semantic centrality based automated requirements prioritization, Proceedings of the 7th India Software Engineering Conference, ACM, 2014, p. 5.
[80] S.I. Mohamed, I. ElMaddah, A.M. Wahba, Towards value-based requirements prioritization for software product management, Int. J. Softw. Eng. 1 (2) (2008) 35–48.
[81] P. Mongeon, A. Paul-Hus, The journal coverage of web of science and scopus: a comparative analysis, Scientometrics 106 (1) (2016) 213–228.
[82] I. Morales-Ramirez, D. Munante, F. Kifetew, A. Perini, A. Susi, A. Siena, Exploiting user feedback in tool-supported multi-criteria requirements prioritization, International Conference on Requirements Engineering Conference (RE'17), IEEE, 2017, pp. 424–429.
[83] D. Mougouei, D.M. Powers, Modeling and selection of interdependent software requirements using fuzzy graphs, Int. J. Fuzzy Syst. 19 (6) (2017) 1812–1828.
[84] B.A. Mustafa, A. Zainuddin, An experimental design to compare software requirements prioritization techniques, International Conference on Computational Science and Technology (ICCST), IEEE, 2014, pp. 1–5.
[85] S. Nidhra, L.P.K. Satish, V.S. Ethiraj, Analytical hierarchy process issues and mitigation strategy for large number of requirements, Sixth International Conference on Software Engineering (CONSEG), IEEE, 2012, pp. 1–8.
[86] C.E. Otero, E. Dell, A. Qureshi, L.D. Otero, A quality-based requirement prioritization framework using binary inputs, 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, IEEE, 2010, pp. 187–192.
[87] S. Parthasarathy, M. Daneva, An approach to estimation of degree of customization for erp projects using prioritized requirements, J. Syst. Softw. 117 (2016) 471–487.
[88] M. Pergher, B. Rossi, Requirements prioritization in software engineering: a systematic mapping study, 2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE), IEEE, 2013, pp. 40–44.
[89] A. Perini, F. Ricca, A. Susi, Tool-supported requirements prioritization: comparing the ahp and cbrank methods, Inf. Softw. Technol. 51 (6) (2009) 1021–1032.
[90] A. Perini, A. Susi, P. Avesani, A machine learning approach to software requirements prioritization, IEEE Trans. Softw. Eng. 39 (4) (2013) 445–461.
[91] Z. Racheva, M. Daneva, Reprioritizing the requirements in agile software development: towards a conceptual model from clients' perspective, International Conference on Software Engineering and Knowledge Engineering (SEKE) (2009) 73–80.
[92] Z. Racheva, M. Daneva, L. Buglione, Supporting the dynamic reprioritization of requirements in agile development of software products, Second International Workshop on Software Product Management, IWSPM'08, IEEE, 2008, pp. 49–58.
[93] Z. Racheva, M. Daneva, A. Herrmann, A conceptual model of client-driven agile requirements prioritization: Results of a case study, Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, 2010, p. 39.
[94] Z. Racheva, M. Daneva, K. Sikkel, A. Herrmann, R. Wieringa, Do we know enough about requirements prioritization in agile projects: insights from a case study, International Conference on Requirements Engineering (RE'10), IEEE, 2010, pp. 147–156.
[95] M.S. Rahim, A.E. Chowdhury, S. Das, Rize: A proposed requirements prioritization technique for agile development, 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), IEEE, 2017, pp. 634–637.
[96] M. Ramzan, M.A. Jaffar, M.A. Iqbal, S. Anwar, A.A. Shahid, Value based fuzzy requirement prioritization and its evaluation framework, Fourth International Conference on Innovative Computing, Information and Control (ICICIC), IEEE, 2009, pp. 1464–1468.
[97] M. Ramzan, M.A. Jaffar, A.A. Shahid, Value based intelligent requirement prioritization (virp): expert driven fuzzy logic based prioritization technique, Int. J. Innov. Comput.Inf. Control 7 (3) (2011).
[98] N. Riegel, Model-based prioritization in business-process-driven software development, International Conference on Requirements Engineering Conference (RE'12), IEEE, 2012, pp. 349–352.
[99] N. Riegel, J. Doerr, A systematic literature review of requirements prioritization criteria, International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2015, pp. 300–317.
[100] N.A. Ring, K. Ritchie, L. Mandava, R. Jepson, A guide to synthesising qualitative research for researchers undertaking health technology assessments and systematic reviews(2011).
[101] V. Sachdeva, Requirements prioritization in agile: use of planning poker for maximizing return on investment, Information Technology-New Generations, Springer, 2018, pp. 403–409.
[102] M. Sadiq, A fuzzy set-based approach for the prioritization of stakeholders on the basis of the importance of software requirements, IETE J. Res. 63 (5) (2017) 616–629.
[103] A. Sapunkov, T. Afanasieva, Software for automation of user requirements prioritization, Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis, ACM, 2019, pp. 1–5.
[104] S. Scheibmayr, T. Hildenbrand, M. Geisser, An experimental, tool-based evaluation of requirements prioritization techniques in distributed settings, Americas Conference on Information Systems (AMCIS), Association for Information Systems, 2009, p. 735.
[105] N. Seyff, I. Todoran, K. Caluser, L. Singer, M. Glinz, Using popular social network sites to support requirements elicitation, prioritization and negotiation, J. Internet Serv. Appl. 6 (1) (2015) 1.
[106] F. Shao, R. Peng, H. Lai, B. Wang, Drank: a semi-automated requirements prioritization method based on preferences and dependencies, J. Syst. Softw. 126 (2017) 141–156.
[107] P. Shao, Sample selection: an algorithm for requirements prioritization, Proceedings of the 46th Annual Southeast Regional Conference on XX, ACM, 2008, pp. 525–526.
[108] Y.V. Singh, B. Kumar, S. Chand, J. Kumar, A comparative analysis and proposing 'ann fuzzy ahp model'for requirements prioritization, Int. J. Inf. Technol. Comput. Sci. 10 (4) (2018) 55–65.
[109] M. Svahnberg, A. Karasira, A study on the importance of order in requirements prioritisation, Third International Workshop on Software Product Management (IWSPM), IEEE, 2009, pp. 35–41.
[110] R.B. Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni, R. Feldt, A. Aurum, Prioritization of quality requirements: State of practice in eleven companies, International Conference on Requirements Engineering Conference (RE'11), IEEE, 2011, pp. 69–78.
[111] R. Thakurta, A framework for prioritization of quality requirements for inclusion in a software project, Softw. Qual. J. 21 (4) (2013) 573–597.
[112] P. Tonella, A. Susi, F. Palma, Interactive requirements prioritization using a genetic algorithm, Inf. Softw. Technol. 55 (1) (2013) 173–187.
[113] G. Vibha, S. Anuja, Identifying an Appropriate Requirements Prioritization Methodology Using Fuzzy Decision-making, Computer Networks and Intelligent Computing, Springer, 2011, pp. 258–268.
[114] P. Voola, A.V. Babu, Requirements uncertainty prioritization approach: a novel approach for requirements prioritization, Softw. Eng. Int. J. (SEIJ) 2 (2012) 37–49.
[115] P. Voola, A.V. Babu, Comparison of requirements prioritization techniques employing different scales of measurement, ACM SIGSOFT Softw. Eng. Not. 38 (4) (2013) 1–10.
[116] R. Wieringa, M. Daneva, Six strategies for generalizing software engineering

theories, Sci. Comput. Program. 101 (2015) 136–152.
[117] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requir. Eng. 11 (1) (2006) 102–107.
[118] C. Wohlin, M. Höst, K. Henningsson, Empirical research methods in software engineering, Empirical Methods and Studies in Software Engineering, Springer, 2003, pp. 7–23.
[119] M. Yaseen, N. Ibrahim, A. Mustapha, Requirements prioritization and using iteration model for successful implementation of requirements, Int. J Adv. Comput. Sci. Appl. 10 (1) (2019) 121–127.
[120] L. Zhang, F. Wang, L. Wang, W. Wang, B. Liu, J. Liu, M. Chen, Q. He, Y. Liao, X. Yu, et al., Prevalence of chronic kidney disease in china: a cross-sectional survey, Lancet 379 (9818) (2012) 815–822.
[121] P. Zheng, X. Xu, S.Q. Xie, A weighted interval rough number based method to determine relative importance ratings of customer requirements in qfd product planning, J. Intell. Manuf. 30 (1) (2019) 3–16.
[122] S.H. Zyoud, D. Fuchs-Hanusch, A bibliometric-based survey on ahp and topsis techniques, Expert Syst. Appl. 78 (2017) 158–181.