Understanding requirements dependency in requirements prioritization: a systematic literature review

Fiftin Noviyanto [a,b,1,*], Rozilawati Razali [a,2], Mohd Zakree Ahmad Nazri [c,3]

[a] Center for Software Technology and Management, FTSM, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
[b] Informatics Department, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[c] Center for Artificial Intelligence and Technology, FTSM, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
[1] fiftin.noviyanto@tif.uad.ac.id ; [2] rozilawati@ukm.edu.my; [3] zakree@ukm.edu.my
* corresponding author

ARTICLE INFO

Article history
Received February 24, 2023
Revised May 7, 2023
Accepted May 16, 2023
Available online June 1, 2023

Keywords
Requirement prioritization
Requirements dependencies
Systematic literature review
Machine learning
Optimization technique

ABSTRACT

Requirement prioritization (RP) is a crucial task in managing requirements as it determines the order of implementation and, thus, the delivery of a software system. Improper RP may cause software project failures due to over budget and schedule as well as a low-quality product. Several factors influence RP. One of which is requirements dependency. Handling inappropriate handling of requirements dependencies can lead to software development failures. If a requirement that serves as a prerequisite for other requirements is given low priority, it affects the overall project completion time. Despite its importance, little is known about requirements dependency in RP, particularly its impacts, types, and techniques. This study, therefore, aims to understand the phenomenon by analyzing the existing literature. It addresses three objectives, namely, to investigate the impacts of requirements dependency on RP, to identify different types of requirements dependency, and to discover the techniques used for requirements dependency problems in RP. To fulfill the objectives, this study adopts the Systematic Literature Review (SLR) method. Applying the SLR protocol, this study selected forty primary articles, which comprise 58% journal papers, 32% conference proceedings, and 10% book sections. The results of data synthesis indicate that requirements dependency has significant impacts on RP, and there are a number of requirements dependency types as well as techniques for addressing requirements dependency problems in RP. This research discovered various techniques employed, including the use of Graphs for RD visualization, Machine Learning for handling large-scale RP, decision making for multi-criteria handling, and optimization techniques utilizing evolutionary algorithms. The study also reveals that the existing techniques have encountered serious limitations in terms of scalability, time consumption, interdependencies of requirements, and limited types of requirement dependencies.

1. Introduction

A large-scale software system development normally involves vast amounts of requirements, which contribute significantly to the success of the system. Software projects, on the other hand, do have resource constraints that impede them from realizing all the requirements at once. In essence, implementing massive requirements with limited resources due to budget, schedule [1], and staff constraints is troublesome [2]. One possible management strategy to resolve the issue is through prioritization. Requirements prioritization (RP) is an essential part of requirements management, aimed at selecting the requirements based on certain predetermined criteria so that they could be implemented in stages [3]–[5]. Several common criteria that determine RP include stakeholders [6], system functionality [7], cost [8], processing time [8], risk considerations [9], [10], and business values [4].

RP activities always involve two parties, namely developers and stakeholders. The two parties have different focuses and thus, priorities [11]. Stakeholders focus on urgency, needs, and business values [12], [13]. Although developers are concerned about project attributes such as effort [12] and cost [14], [13], they are also aware of internal constraints such as dependency between functions or requirements [12]. This is due to the fact that requirements dependency is commonly found on the project software [2], [15]. It is thus risky to conduct RP without considering the dependency between requirements [16], [17]. For instance, giving high priority to requirements that depend on other requirements can increase the waiting time and delay the project [18]. This is because the dependent requirements have to wait for the prerequisite requirements to be completed before they could be implemented. In addition, requirements dependency also implies product complexity [2] and project risk [19]. The higher the dependency, the higher the complexity of the system and thus the higher its risk of failure is [20].

Several studies have investigated RP concerning the criteria and techniques used in the process, such as Hujainah et al. [21], Tan and Mohamed [22], Falak Sher et al. [23], Muhammad Sufian et al. [24], Pitangueira et al. [25], Achimugu et al. [16], and Al Ta'ani and Razali [26]. However, none of the studies examine requirements dependency in depth. In fact, only 4 out of the 65 RP techniques consider requirements dependency [2]. Many studies on RP do not include requirements dependencies as one of the factors influencing priority sequencing. Little is known about requirements dependency in RP. Therefore, this study aims to explore further requirements dependency in RP by conducting a systematic literature review (SLR) in order to improve the understanding of the phenomenon. The objectives are:

- to investigate the impacts of requirements dependency on RP
- to identify different types of requirements dependency
- to discover the techniques used for requirements dependency problems in RP

The structure of the paper is as follows. Section 2 describes the methodology used in the review. Section 3 discusses the threats to validity. Section 4 presents the results and discussions. Section 5 concludes the study.

2. Method

This study adopts the Systematic Literature Review (SLR) method proposed by Kitchenham et al. [27]. Fig. 1 illustrates the review protocol used, which comprises five stages: identification, search strategy, study selection strategy, data retrieval, and result.

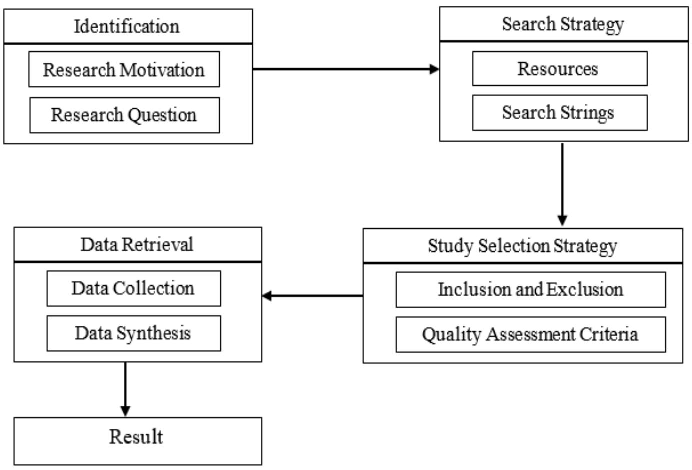

Fig. 1. Review Protocol

In the first stage, the research questions were constructed and aligned with the research motivation and research question. The second stage determined the resources and the search strings based on the research questions. The third stage outlined the inclusion and exclusion criteria for screening the gathered articles together with the Quality Assessment Criteria (QAC) process. The fourth stage finalized the selection of the data collection through which data synthesis was made. In the final step, the results of the synthesis were obtained, which are presented in this paper.

2.1. Research Questions

The study aims to understand the relationships between requirements dependency and RP. Therefore, the following research questions (RQ) were constructed:

- RQ1: Does requirements dependency have impacts on RP?
- RQ2: What are the different types of requirements dependency?
- RQ3: What are the existing techniques used for requirements dependency problems in RP?

2.2. Search Strategy

The search strategy undertaken in this study began with determining the sources of scientific literature. Seven sources were used in the literature search, as listed in Table 1.

Table 1. Selected Literature Database Resources

Resource Name	Resource Link
SpringerLink	http://www.springerlink.com
Google Scholar	https://scholar.google.com
ISI Web of Knowledge	http://www.isiknowledge.com
Elsevier	http://www.elsevier.com
ScienceDirect	http://www.sciencedirect.com/
IEEE Xplore	http://www.ieee.org/web/publications/xplore/
ACM Digital Library	http://portal.acm.org

- Resources: The seven sources were selected because their contents are relevant to the subjects of this study, besides being referred by researchers in the field.
- Search Strings: Specifically, the keywords used for searching research articles in this study were 'requirement prioritization' or 'dependencies'. The search keywords for review papers were 'requirement prioritization' (AND/OR) 'literature review'.

2.3. Study Selection Criteria

The searches in the seven sources using the predefined keywords found 432 articles. These articles were firstly screened in terms of suitability based on their titles and/or abstracts. As a result, only 133 articles were selected. Fig. 2 shows the distribution of the articles. Most articles are journal and conference papers.

To ensure only the most relevant articles would be selected, the 133 articles were further vetted through several subsequent stages, as shown in Fig. 3. Inclusion and Exclusion Criteria: The inclusion criteria used to select the articles are as follows: 1) Articles are written in English; 2) Articles focus on requirements dependency in RP domain; and 3) Articles are able to answer at least one of the research questions. The exclusion criteria include: 1) Articles are not written in English; 2) Duplicate articles - excluding multiple copies of the same study; and 3) Articles are not answering any of the research questions. Each collected article was briefly read through its title, abstract, and content. Studies that did not address the research question were excluded. Similarly, studies that were still in the research process or not published by a publisher were not included. This study aims to gather findings that have been proven empirically. Therefore, review articles were excluded from the selection. If the same article was found from different sources, only one would be chosen. The articles were published within the period of 2012 to 2022.

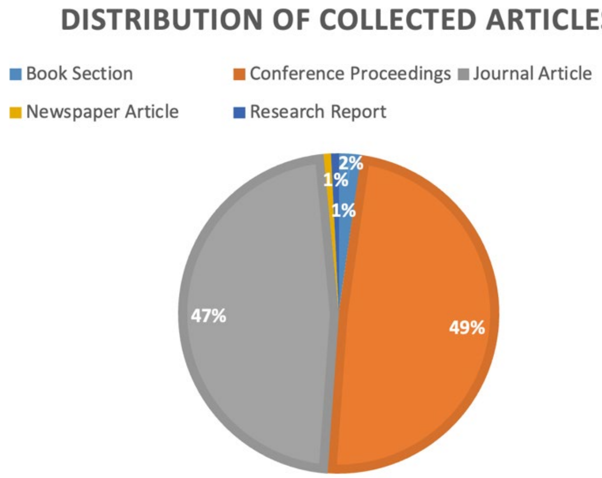

Fig. 2. Precentage of collected articels based on titles and/or abstracts

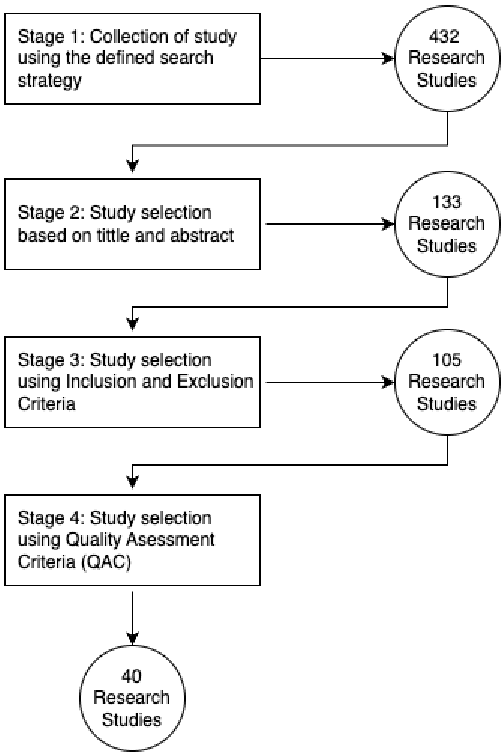

Fig. 3. Precentage of collected articels based on titles and/or abstracts

Quality Assessment Criteria: Quality Assessment Criteria (QAC) was used to measure the quality of the gathered articles with respect to the objectives of the study. First, the articles shall cover requirements dependency and RP. Second, the articles shall be trustworthy. Eight questions were derived to represent the criteria, as listed in Table 2. The possible score for each question was divided into three: Yes (1), Partially (0.5) and No (0). The weighted score for each study was the sum of scores for the eight questions. The assessments were conducted by the authors, through which the scores were consensually determined.

Table 2. Question for Quality Assessment Criteria

ID	Question (Q)	Answer Score
Q1	Is the objective of the research related to requirements prioritization clearly stated?	Yes = 1/ partially= 0.5/no = 0
Q2	Does the study focus on the requirement prioritisation?	Yes = 1/ partially= 0.5/no = 0
Q3	Does the study focus on the dependences on RP?	Yes = 1/ partially= 0.5/no = 0
Q4	Does the study illustrate the current various/complexity/types of requirement dependencies on RP?	Yes = 1/ partially= 0.5/no = 0
Q5	Does the study explain the proposed techniques to handle requirements dependencies in RP?	Yes = 1/ partially= 0.5/no = 0
Q6	Are the measures used in the study the most relevant ones for answering the research questions (this study)?	Yes = 1/ partially= 0.5/no = 0
Q7	Does the research contribute to requirements prioritization considering requirements dependencies?	Yes = 1/ partially= 0.5/no = 0
Q8	Is the result of the study clearly stated?	Yes = 1/ partially= 0.5/no = 0

After the assessment, only forty articles were selected based on the weighted score > 4.5. A score of 4.5 was used as the baseline as it designates that the article has achieved more than 56% of the best score (4.5 out of 8). Table 3 presents the weighted scores for the forty selected articles. The highest score is 8 (five articles) and the lowest score is 5 (five article), whereas the median score is 6 (twelve articles). This implies that the selected articles are well-documented, and thus contain well-conducted studies. This claim is particularly demonstrated by the scores attained for Q1, Q7 and Q8, which are mainly 1 (Yes). The selected articles are however moderately covering the focus of this study, as the scores for Q2 to Q6 are mostly 0.5 (Partly). This indicates that the emphasis of the current available studies on requirements dependency and RP are still lacking, albeit relevant. Only five articles (Score = 8) fulfill the quality criteria of the study entirely.

Table 3. Quality Assessment criteria results

Reference of the selected study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Score
[7]	1	1	0.5	0.5	0.5	0.5	1	1	6
[18]	1	1	1	1	1	1	1	1	8
[28]	1	1	1	1	1	1	1	1	8
[29]	1	1	1	0.5	1	1	1	1	7.5
[30]	1	1	0.5	0.5	0.5	0.5	1	1	6
[31]	0.5	1	1	0.5	1	1	1	0.5	6.5
[32]	1	1	0.5	0	0.5	0.5	1	1	5.5
[12]	1	1	0.5	1	0.5	0.5	1	1	6.5
[33]	1	1	0.5	0.5	0.5	1	1	1	6.5
[34]	1	0.5	0.5	0.5	0.5	0.5	1	1	5.5
[35]	1	1	1	0.5	1	1	1	1	7.5
[36]	0.5	1	0.5	0.5	0.5	0.5	1	1	5.5
[26]	1	1	0.5	0.5	0.5	0.5	1	1	6
[37]	1	1	0.5	0.5	0.5	0.5	1	1	6
[38]	1	1	0.5	0.5	0.5	0.5	1	1	6
[39]	1	0.5	0.5	0.5	0.5	0.5	1	1	5.5
[40]	1	1	1	1	1	0.5	0.5	0.5	6.5
[41]	1	1	0.5	0.5	0.5	0.5	1	1	6
[42]	1	0.5	0.5	0.5	1	0.5	1	1	6
[43]	1	1	1	0.5	1	1	1	1	7.5
[44]	1	1	1	0.5	0.5	0.5	1	1	6.5
[45]	1	1	1	1	1	0.5	1	1	7.5
[46]	1	0.5	1	0.5	0.5	0.5	1	1	6
[47]	1	1	0.5	0.5	0.5	0.5	1	1	6
[48]	1	1	1	1	1	1	1	0.5	7.5
[49]	1	1	1	1	1	1	1	1	8
[50]	1	1	0.5	0.5	0.5	0.5	1	1	6
[51]	1	1	0.5	0.5	1	1	1	1	7
[52]	1	1	0.5	0.5	0.5	0.5	1	1	6

Reference of the selected study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Score
[53]	1	0.5	0.5	0.5	0	0.5	1	1	5
[54]	1	0,5	0,5	0	0	1	1	1	5
[55]	1	0,5	1	1	1	1	1	1	7,5
[56]	1	1	0	0	1	1	1	1	6
[57]	1	1	1	1	1	1	1	1	8
[58]	1	1	1	1	1	1	1	1	8
[59]	1	1	0	0	0	1	1	1	5
[60]	1	0,5	1	1	1	1	1	1	7,5
[61]	1	1	0	0	0	1	1	1	5
[62]	1	1	0	0	0	1	1	1	5
[63]	1	0,5	1	1	1	1	1	1	7,5

2.4. Data Retrieval

The data retrieval consists of two activities, namely data collection and data synthesis. Data collection is the process of bringing together the selected articles, whereas data synthesis is a purposeful activity that extracts facts from the selected studies for answering the stated research questions [27].

- Data Collection: This stage gathered and consolidated the selected thirty articles. The articles were then classified into three groups based on the three RQs: RQ1, RQ2 and RQ3. For example, the articles that discuss the impact of requirements dependency on RP were placed under RQ1 group. Same goes to the articles that belong to RQ2 and RQ3. The articles that address more than one RQ were placed accordingly into the respective RQ groups.
- Data Synthesis: This stage extracted facts from the grouped articles in order to find the answers for the research questions. The facts were then analyzed and visualized. For example, the findings for RQ1 are presented as a chart that shows the frequency distribution of articles across RP factors. Similarly, the requirements dependency types for RQ2 are illustrated as a taxonomy graph, whereas the techniques for RQ3 are demonstrated as chart and table. The visualization helps in explaining the results, thus providing a better understanding of the phenomenon.

3. Threats of validity

The main challenge in SLR is the validity of the study, which includes the completeness, publication bias and data synthesis [27]. This study adopted the review protocol to overcome the completeness threat. The searches were conducted on various databases and the articles were screened using the predetermined quality criteria. Nevertheless, the searches were limited to publications from year 2012-2022 and articles in other languages were excluded. The consideration for choosing English is due to its status as an international language widely used in reputable journals. To avoid publication bias, only articles that contain empirically proven data were considered. Therefore, gray studies that are still in progress were not included. The consideration to exclude gray literature is the ease of literature search for future researchers. To mitigate the data synthesis threat, QAC process was conducted. QAC identified and filtered reliable studies that could answer the research questions. Moreover, manual checks were carried out on the extracted facts repetitively. The assessments were carried out objectively and consensually by the authors to avoid inconsistencies. The authors read the entire collected papers and provided scores based on the QAC questions.

4. Results and Discussion

This section describes the results of the analysis based on the forty selected articles.

4.1. Overview of Selected Primary Research Studies

Fig. 2 shows that the most selected articles are journal and conference papers (96%), whereas the rest are book sections, newspaper articles and reports (4%). After QAC process, the distribution changes slightly as shown in Fig. 4. Most articles still constitute journal and conference papers (92.5%), while the rest are book sections only (7.5%). As newspaper articles and reports generally lack scientific evidence and arguments, they could not be selected in this round.

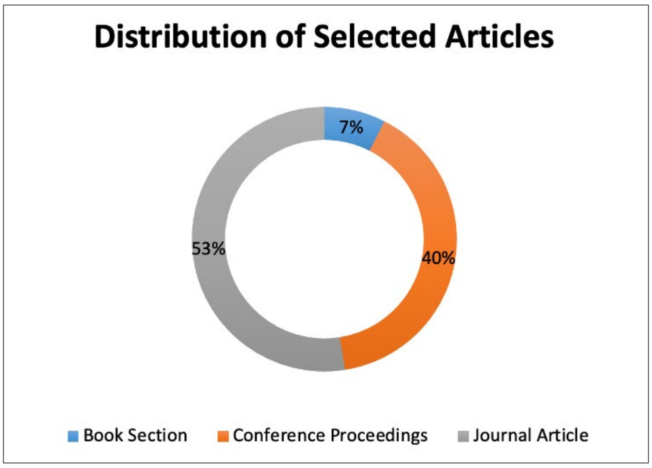

Fig. 4. Percentage of selected articles

Fig. 5 displays the distribution of the selected articles within the period of 2012 until 2022. The chart shows the topics which are consistently studied every year for the last eleven years, with at least three articles per year (median). The number is not high, this indicates that requirements dependency and RP are two topics investigated by the research community in recent years. Since it is not as many as other topics in requirements engineering field, this may suggest that more studies are required to investigate the topics.

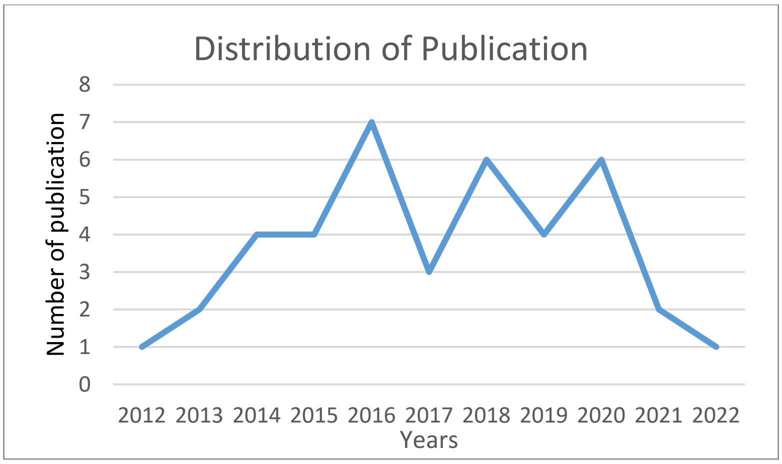

Fig. 5. Number of selected articels by publication year

4.2. Does requirements dependency have impacts on RP (RQ1)?

Fig. 6 indicates that requirements dependency is a critical factor that is of concern to many studies in regards to RP. The articles emphasize that requirements dependency becomes more challenging, particularly for large-scale systems [28], [35], [48]. Improper handling of requirements dependency may cause inefficiency [29], project delays [54], redesign and rework [52], as dependencies among requirements are commonly found in software projects [64]. If a requirement that becomes a prerequisite to other requirements is given a low priority, it affects the completion time of the whole project [18], [40]. The prerequisite requirements, therefore, need to be given a higher priority. This implies that requirements dependency determine the complexity of relationships between requirements and thus contributes to erroneous or redundant results [30] and also implies a higher requirement implementation risk [44], [55]. In the requirement prioritization process, there are two different perspectives from the stakeholders and developers. On the client-side, priorities depend on urgency, needs, and business value.

On the developer side, the priority is influenced by something more technical in the system development process, which is the requirements dependencies. [29], [36]. The results of the qualitative research conducted by Al Ta'ani [26] obtained the same result, indicating that analysts and system developers considered dependency as an important factor in requirement prioritization.

Fig. 6 shows that cost and risk are also relatively significant in RP. In general, cost and risk are implicitly influenced by the complexity of requirements, among others. The higher the complexity, the higher the cost and the risk of implementing the requirements are [20]. As discussed earlier, requirements dependency causes requirements complexity [2]. This fact indirectly highlights further the impacts of requirements dependency on RP.

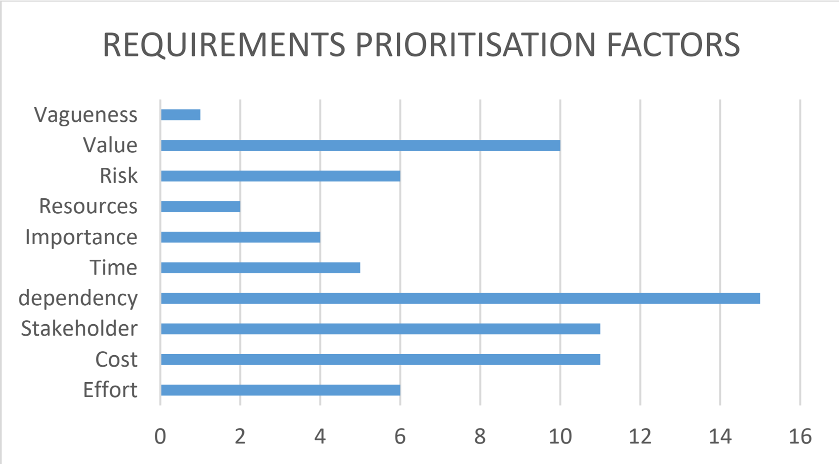

Fig. 6. Frequency of requirements prioritisation factors

4.3. What are the different types of requirements depenceny?(RQ2)

RQ2 focuses on extracting the types of requirements dependency. In general, there are two main classifications of requirements dependency proposed by [65] and [66]. As illustrated in Table 4, the former classifies dependency into three groups [65]: Functional; Value-related, and Time-related. The Functional consists of Combination, Implication, and Exclusion. Combination refers to the requirements to be implemented together and Implication is the requirements that must wait for other requirements to complete. Exclusion is the opposite of Combination, comprising the requirements that cannot be applied together as they are conflicting with each other. On the other hand, Value-related consists of Revenue-based and Cost-based; Revenue-based are requirements that can affect income, whereas Cost-based are requirements that can affect costs. The last group is Time-related, which is requirements that need to be implemented based on the time stated in the project schedule.

Table 4. Dependency Classification Based On C. Li

Dependency group	Dependency Type
Functional dependency	Combination Implication Exclusion
Value-related dependency	Revenue-based Cost-based
Time-related dependency	Time-related

The latter classifies dependency into three types, as shown in Table 5, namely Structural Interdependencies, Constrain Interdependencies, and Cost/Value Interdependencies [66]. Structural Interdependencies consist of four: Refined to; Change to; Similar to; and Requires. Constrain Dependencies consist of Requires and Conflicts with. The Requires are included in both Structural and

Constrain Interdependencies. Another classification is Cost/Value Interdependencies, which comprise Increase/Decrease Cost of and Increase/Decrease Value of.

Table 5. Dependency Classification Based On Dahlstedt's Model

Dependency group	Dependency Type
Structural Interdependencies	Refined_to Change_to Similar_to
Constrain Interdependencies	Requires Cost-based Conflicts_with
Cost/Value Interdependencies	Increases/ Decreases_ cost_of Increases/ Decreases_ value_of
Structural Interdependencies	Refined_to

In addition to the above classifications, there are also articles that mention indirectly and solely other types of requirements dependency. The articles mostly use different terms, even though they refer to the same kinds of dependency. In order to avoid redundancy and inconsistency, this study joins similar types together and assigns coherent terms that represent the classifications best. Fig. 7 illustrates the taxonomy view of requirements dependency categories synthesized from various classifications proposed in the selected articles.

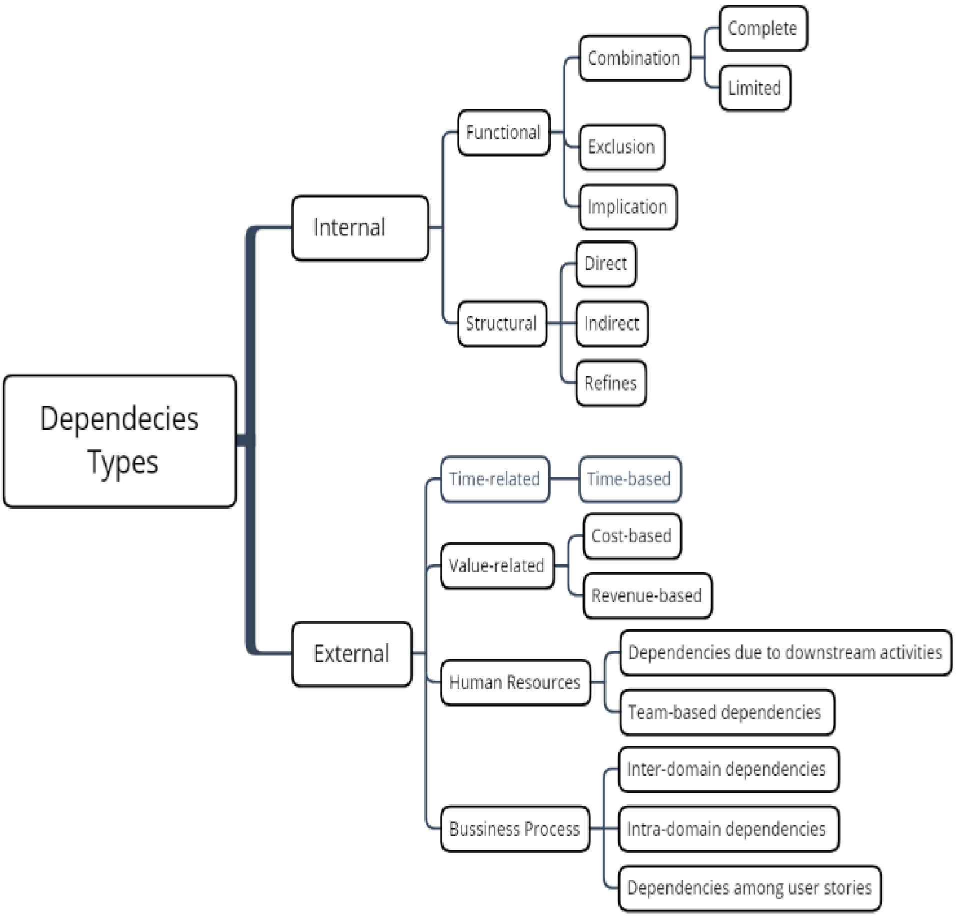

Fig. 7. Types of requirements dependency

Overall, there are two requirements dependency categories: Internal and External. Internal dependency means interior attributes of the system that cause its requirements interdependencies. External dependency means exterior attributes that affect or influence the requirements of the system. Internal dependency has two subcategories, namely, Functional and Structural. External dependency is divided into four subcategories which consist of Time-related, Value-related, Human Resource, and Business Process.

For the Functional subcategory in Internal category, there are three types of dependency including:

- *Combination* [42], [65], [66] is a pair of requirements that must be applied together. Other similar terms used are Coupling [7], Concurrence [35], Requires [54], [55], and Constrain [18], [49]. This type has two subtypes, namely Complete [29], and Limited [29]. Complete is dependent on another requirement completely while Limited is partly dependent on another requirement.
- *Exclusion* [42] is a pair of conflicting requirements, which cannot be applied together. Other similar terms used are Conflicts [35], Contradict [43].
- *Implication* [42] is a requirement that requires other requirements to function. Other terms used are Precedence [7], [51], [58], Time-related [42], [65] and Support [43].

Likewise, there are three types for the Structural [66] subcategory, namely:

- Direct [29], [41] means that requirements depend directly on other requirements. For example, X depends on Y directly.
- Indirect [29] means that requirements depend on other requirements indirectly. For example, X depends on Z, while Z depends on Y. This shows X depends on Y but through Z.
- Refines [35], [67] means that requirements of higher levels are explained by a number of requirements of lower levels. Another term used for this type is hierarchy [31].

On the other hand, the External category consists of Time-related, Value-related, Human resources, and Business processes. There is only one type of Time-related sub-category, namely *Time-based* [42], [65]. This means a requirement that needs to be implemented based on the time stated in the project schedule. Meanwhile, in the Value-related subcategory, the two types comprise:

- Cost-based [28], [42], [65], [58], [57], means a requirement that can affect cost. Other terms found are Contribution [35] and Cost-related [58].
- Revenue-based [42], [47], [65] means a requirement that can affect income.

There are two types of Human resource subcategory, namely:

- Dependencies due to Downstream Activities [52] imply requirements whose implementation considers optimizing existing human resources.
- Team-based Dependencies [52] concern about avoiding multiple teams having to work on the same or on dependent requirements.
- The last sub-category of External category is the business process, which has three types as follows:
- Inter-domain Dependencies [52] indicate requirements whose implementation depends on requirements across business sectors.
- Intra-domain Dependencies [52] indicate requirements whose implementation depends on certain business processes.
- Dependencies among user stories [52] indicate dependencies between non-functional requirements (e.g. usability, maintainability) and architecture choices.

4.4. What are the existing techniques used for requirements dependency problems in RP? (RQ3)

There are various techniques proposed in the selected articles for solving requirements dependency problems in RP. All the techniques used in the selected studies (based on QAC) were analyzed, clustered, and studied in their process. In general, the discovered techniques have specific problem criteria. Decision Making is used to address RP with multiple criteria: Evolutionary Algorithm for computational optimization, Fuzzy logic to handle uncertainty factors, NLP for automated identification of RP and RD based on human language, Machine Learning for automatic determination of RP based on datasets, and Graph-based approaches for mapping RD within groups of requirements.

The most commonly used techniques found in the selected articles are Decision Making, including Collaborative requirement prioritization method [12], Utility-based prioritization [68], Majority Voting Goal-Based (MVGB) [37], Analytic Hierarchy Process (AHP) [18], [50] and Hierarchical Dependencies [31]. One of the Multi-Criteria-Decision-Making techniques is AHP. AHP has excellent accuracy since pairwise comparison is able to provide decisions that are accurate and worth considering [69]. However, pairwise comparison is time-consuming for large scale projects [37].

The second-highest technique is Evolutionary Algorithm (EA), which comprises the Least-Squares-Based Random Genetic Algorithm [30], Hybrid Enriched Genetic Revamped Integer Linear Programming [42], Multi-objective Evolutionary Algorithms (MOEAs) [15], MOSAs [58], Interactive Genetic Algorithm (IGA) [51] and Early Mutation Testing [32]. The most widely used EA technique is the Genetic Algorithm (GA). This technique aims to reduce computation time. It can be combined with other techniques that are able to provide better accuracy. In the selected articles, EA is only used in simulation cases. Thus, it needs to be proven in industry settings.

The next category is Fuzzy Logic. There are three techniques in this category, namely the Hierarchical Fuzzy Inference System (HFIS) [7], Fuzzy Inference System (FIS) [45], Fuzzy Clustering [62], Rough Set Theory [63], and Tensor and Fuzzy Graphs [28]. Fuzzy is used to help in the decision-making process. Each stakeholder's perception of the value of a requirement is different, which is mainly based on interests and knowledge. Fuzzy Logic can be used to solve uncertainty problems due to human judgment.

Previous studies also use Neuro-Linguistic Programming (NLP) for requirements dependency in RP, such as Satisfiability Modulo Theories (SMT) [48] and SNIPR [70]. SNIPR completes SMT. NLP is used as the input for both techniques. Requirements are clustered using NLP and combined with weighting dependencies. The ranking process is combined with GA [48] and AHP [70]. NLP is quite helpful in filtering requirements, thus minimizing redundancy and similarity. Nevertheless, NLP still needs to be explored more in detecting dependencies between requirements, so that costs and time can be further optimized, especially for large scale projects.

Other existing techniques are Graph and Matrix. They are used to visualize and calculate relation weights. The Matrix can be applied separately [34], [71], [44], [38] or in conjunction with Graph [43], [57]. Graphs are composed of nodes, which represent requirements, and edges as relations. Matrix, on the other hand, consists of rows and columns, with cells showing relations between requirements. Because of the visual representation, both techniques make it easy to view dependencies among requirements. However, the techniques would consume time and cost for large-scale requirements.

Machine learning (ML) has been introduced to automate the process of RP. There are five ML methods for requirements dependency in RP, namely CDBR [29], DRank [35], Active Learning [60], Supervised Classification Technique [55], and Interactive Next Release Problem (iNRP) [39]. First, CDBR exploits the Particle Swarm Optimization (PSO) method [29]. The technique minimizes conflicts between stakeholders and developers using a variety of population sizes, between 10 to 50 set requirements. On the computational time and complexity side, CDBR shows excellent results compared to AHP. Second, DRank uses the RankBoost algorithm for learning and calculating requirements dependencies [35]. The graph is used to show or represent dependencies between requirements. There are two types of a graph generated by the DRank method—the first graph is for representing contributions, and the second graph is to represent business rules. Third, iNRP uses Least Median

Square (LMS) and Multilayer Perceptron (MLP) techniques [39]. Time-consuming testing, placing DRANK, is superior to the AHP and CBRank methods [35]. In general, ML could be used to reduce interactions with practitioners. It provides better computational efficiency at significant scalability. However, the challenge is the availability of datasets and the selection of techniques that fit the project's characteristics.

Fig. 8 shows the distribution of techniques used to address requirements dependency problems in RP, based on their technical bases. Most techniques seem to employ Decision Making and Evolutionary Algorithm technical bases.

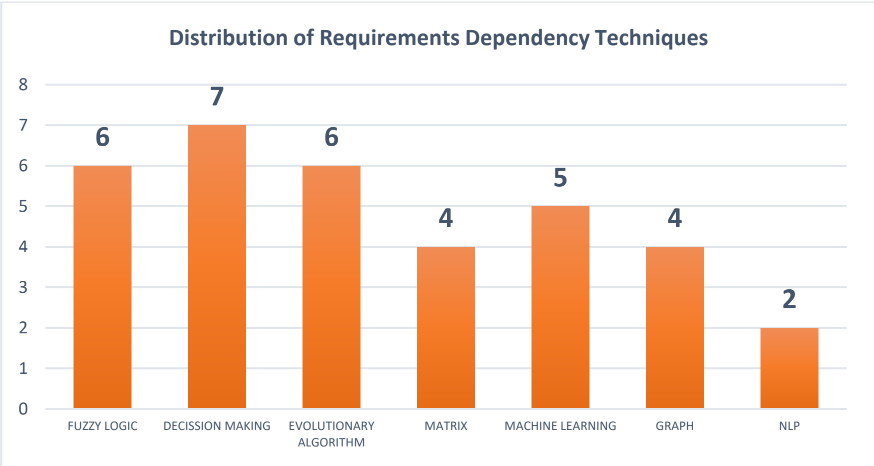

Fig. 8. Techniques of handling requirements dependency in requirements prioritisation

The brief explanation of each technique in terms of process and limitations is presented in Table 6. Based on the synthesis presented in the table, each technique has its own designated approaches in solving requirements dependency problems in RP. It can be seen that most techniques to date only handle small numbers of requirements and cover limited types of dependency. They in fact have yet to be tested using real cases with large data sets. Some are incomplete and shallow, covering trivial aspects of the matter. Stakeholders have different perceptions in assessing the importance of requirements due to diverse backgrounds and knowledge. The Fuzzy Logic techniques are widely used to solve the problem of uncertainties among stakeholders in determining priorities [7], [28], [45].

Decision Making techniques help RP by comparing requirements [12], [18], [31], [33], [37], [50], and [54]. The techniques are accurate, but the comparisons become complex when they involve a large number of requirements. The techniques therefore are not suitable for large-scale projects. ML can help resolving the issue of large data. Prior to that, ML however requires training data that are based on the generated knowledge base containing patterns or rules. Graph [38], [40], [43] and Matrix [34], [43], [44], [71] can aid to visualize the dependencies between requirements. As these techniques only focus on visualization, they usually have to be combined with other techniques because RP needs to consider the values of interest between requirements.

NLP works by reading and recognizing patterns. Although promising, a common obstacle of applying this technique is the inconsistency in the written requirements [48], [70]. As such, reading the patterns is difficult. Recognizing discrete patterns is also not so straightforward, as dependencies vary. EA is a metaheuristics-based technique [30], [42], [47], [51], [72]. Its advantage is the gene selection, which means only the best requirements can survive. EA can be combined with another method that exploits a genetic algorithm to reduce the number of pairs of elicited requirements [51]. Only pairs that allow the disambiguation of equally ranked or differently ranked requirements are elicited. However, this technique has yet to be applied in real projects to truly prove its practicality.

Table 6. Process and Limitation of Requirements Dependencies Techniques

Technique	Description	Process	Limitation
Hierarchical Fuzzy Inference System (HFIS) [7] Technical base: FUZZY LOGIC	HFIS is built hierarchically and deals with uncertainty about human judgment.	• Data collection • Preprocessing (include Dependencies) • Ranking uses HFIS • Release plan generation	It only handles two types of dependency: Coupling (Combination) and Precedence (Implication).
AHP [18] Technical base: DECISION MAKING	AHP-based prioritisation is performed pairwise by comparing every requirement against each other.	• Matrix and start comparing all requirements • Pairwise comparison of requirements • Obtain values for each requirement • Sum the row for each column • Divide each value by sum the rows • Averaging and normalization • Priority out of 1 or 100	Cannot be applied efficiently for large size requirements
Tensor and Fuzzy Graphs [28] Technical base: FUZZY LOGIC	Fuzzy-Graph is defined: its nodes are set of nonempty identified requirements R = {r1,..., rn} and the edges show the explicit cost relation among the requirements as C = R ×R Tensor Algebra is a multidimensional array that generalizes the representation of the matrix, and each dimension of the tensor is called a mode.	• Eliciting importance values of functional requirements (FRs) regarding nonfunctional requirements (NFRs) • Generating a primary prioritisation list using the tensor concept • Generating a fuzzy graph of "increase/decrease cost of" dependency • Generating order of cost dependency • Integrating prioritization • Final prioritisation	It handles small numbers of requirements - need to be tested in large-scale requirements
Collaborative requirement prioritisation approach (CDBR) [29] Technical base: MACHINE LEARNING	An iterative hybrid approach called the Collaborative Dependency-Based Ranking approach follows a priori and a posterior perspective for requirement prioritisation. CDBR exploits the use of Particle Swarm Optimization (PSO)	• The cause of Initial Priority Assignment • Initial Priority Assignment by stakeholder • Initial Priority Assignment by developer • Final priority estimation using Particle Swarm Optimization	It only handles two types of dependency: Direct and Indirect.
Least-Squares-Based Random Genetic Algorithm [30] Technical base: EVOLUTIONARY ALGORITHM	Improve method from the interactive genetic algorithm. This method begins from select the initial population by analyzing some partial input orders, or it can be done randomly.	• The population is initialized • Maintain crossover mutation • The threshold value should be null • The main loop is entered in the algorithm • Start the while loop in which disagreement is higher than the threshold value • The process which is to be performed first • The result of the comparison is stored • The survey is to be utilized	It handles small numbers of requirements - need to be tested in large-scale requirements
Hierarchical Dependencies [31] Technical base: DECISION MAKING	Modified AHP that consider the relationships between the stakeholders' needs and the derived requirements in the form of use cases and non-functional requirements	• Elicitation of New Requirements • Requirements Integration • Relativeness Computation • Requirements Prioritisation • Rearrangement of Requirements	It handles small numbers of requirements - need to be tested in large-scale requirements

Technique	Description	Process	Limitation
Early mutation testing [32] Technical base: EVOLUTIONARY ALGORITHM	The mutation process is involved in the modification of software artifact (e.g., CS) by injecting artificial faults. Each mutated version is called a mutant.	• Mutant generation using MutML • Evaluation of the test suite adequacy • Ranking of the test suites • Adequacy score selection • Scenarios identification • Mapping test cases to requirements • Dependencies analysis • Prioritisation of requirements	It handles simple and small numbers of requirements - need to be tested in large-scale requirements Manual process
Collaborative requirement prioritisation method [12] Technical base: DECISION MAKING	A collaborative method where both developers and stakeholders are equally involved in assigning final priority.	• Stakeholder's perspective and their priority inputs • Developer's perspective and their priority inputs • Dependency computation • Dependency classification • Requirement weight computation classification • Priority assignment rules to calculate the developer's priority • Prioritisation process	It only handles three types of dependency: Complete, Limited and Inferred
Utility-based prioritisation [68] Technical base: DECISION MAKING	Utility-based prioritisation allows stakeholders to prioritise a requirement with regard to different interest dimensions.	• Contribution of requirements to the interest dimensions • Predefined weights for the interest dimensions • Ranking of requirements with static weights	Need to analyse which features (interest dimensions) are useful to improve prediction quality.
Requirements Change Analysis [34] Technical base: MATRIX	A method based on the changes that change themselves, which are initiated at higher levels.	• Analyzing the change using functions • Identifying the difficult changing and • Identifying the dependencies using a matrix	Need to identify the effort to implement a requirement change and to apply the method to a more complex case study.
DRank [35] Technical base: MACHINE LEARNING	An automated method by combining machine learning with the link analysis technique	• Select ranking criteria • Select a scale value for each requirement • Prioritise sampled requirements pairs • Generate subjective requirements prioritisation • Generate requirement dependency graphs (RDGs) • Analyze contribution order • Integrate the prioritisation	It only handles two types of dependency: Contribution and Business
Enhancing the Process of Requirements Prioritization in Agile Software Development - A Proposed Model [36] Technical base: Technical Scale	A new RP models conducted based on the exiting RP models.	• Requirements on project • Select requirement based on Business Value and Risk • Prioritised project backlog based on Effort Estimation and Dependency • Sprint backlog	A future research is required to validate the proposed improvements for both the model and the technique.
Majority Voting Goal-Based (MVGB) [37] Technical base: DECISION MAKING	Prioritising the requirements with the specific concern of stakeholders.	• Defining evaluation function • Finding dependency level for each requirement • Finding the Requirement Prioritisation Value (RPV) for each requirement • Selecting the requirements by highest RPV	It causes high cost/effort as compared to other techniques.

Technique	Description	Process	Limitation
Software Features Prioritisation [38] Technical base: GRAPH	A model prioritisation based on the node centrality in the probability network	• FPN is generated from an FM according to the dependencies between features. • The centrality values of all nodes in the generated FPN are calculated and regarded as metrics for feature prioritisation.	It supports a small number of features. The technical performance has not been evaluated.
Interactive Next Release Problem (iNRP) [39] Technical base: MACHINE LEARNING	The model composed of three different components with distinct responsibilities: (a) interactive genetic algorithm, (b) interactive module and (c) learning model	• DM to specify two architectural settings • The weight of the implicit preferences in comparison to the explicit ones for the fitness calculation. • A minimum number of interactions in which the DM is willing to take part in. • The learning process is performed using the set of samples collected in the previous stage as a training dataset	The learning model performance has not been evaluated.
Graphs and Integer Programming [40] Technical base: GRAPH & MATH	The integer programming model for requirement selection which maximizes the overall value of selected requirements while mitigating the adverse impact of selection deficiency problem (SDP). Graph-based for capturing the requirements dependencies.	• Input requirement set • Identify requirement dependencies • Model requirement dependencies • Specify the budget range • Run the selection model and run the propose selection model • Compare the result	This study does not use a technique to handle dependencies specifically.
Hidden Structure Method [71] Technical base: MATRIX	A method focused on analyzing a Design Structure Matrix (DSM) based on coupling and modularity theory, and it has been used in a number of software architecture and software portfolio cases.	• Identify the direct dependencies and compute the visibility matrix • Identify and rank cyclic groups	The requirements dependencies have not been weighted.
Hybrid Enriched Genetic Revamped Integer Linear Programming [42] Technical base: EVOLUTIONARY ALGORITHM	Hybrid EGRILP is a combination of Enriched Genetic Algorithm (EGA) with Revamped Integer Linear Programming (RILP) model	• Initial requirements • Group requirements based on dependencies • Fitness value calculation • Algorithm (EM) Algorithm • Aging factor • Revamped Integer Linear Programming (RILP)	It supports a small number of features and time dependency only.
Component-Based Software Development (CBSD) [43] Technical base: MATRIX & GRAPH	CBSD is an approach to software development that relies on the reuse of software components to reduce the development costs and production cycle while increasing the final product's quality.	• Define the type of relation between each pair of functional requirements • Create a two-dimensional matrix where each row represent a functional requirement, and the rows represent the same series of requirements • Generate a directed graph from the support relationships • Apply the preceding relationship • Apply the conflicted relationship	The proposed algorithm has not been tested.
Traceability Metrics [44] Technical base: MATRIX	Methods to provide support for understanding relations between requirements	• Parse artifacts and trace links • Generate traceability graph • Calculate traceability metrics	It focuses on Agile process and risk factor only.

Technique	Description	Process	Limitation
Fuzzy Inference System (FIS) [45] Technical base: FUZZY LOGIC	A mathematical interpretation to model and deal with the uncertainty in human estimation and their limited knowledge.	• Preprocessing • Ranking • Generating the release plan	It handles small numbers of requirements and only supports two types of dependency: Coupling (Combination) and Precedence (Implication).
Multi-objective evolutionary algorithms (MOEAs) [15] Technical base: EVOLUTIONARY ALGORITHM	A framework which is a Java-based for general-purpose multi-objective optimization algorithms.	• Generate dataset • Applying four evolutionary algorithms: NSGA-II, MOEA, GDE3, and MOEA/D • Evaluation	It uses synthetic dataset and supports value-based dependency only.
Satisfiability Modulo Theories (SMT) [48] Technical base: NLP	This method is aimed to obtain a proper initial population.	• Requirement elicitation • Obtain requirements dependencies • Formalization • SMT solver • Genetic algorithm • Requirement prioritised documents	The proposed method has not been tested in industrial setting.
SNIPR [70] Technical base: NLP	Methods for prioritising requirements that exploit NLP in assisting the user in identifying interdependencies and constraints between requirements	• Requirements elicitation • Identify requirement dependencies and priority (NLP) • Rank requirements and identify disagreements (SMT solver) • Rerank the subset of requirements for improved accuracy (AHP) • Ranked and Selected Requirements	Testing is limited to 100 requirements – need to be tested in large-scale requirements
AHP [50] Technical base: DECISION MAKING	A decision-making method that compares all pairs to find a higher priority	• The relationships between the different requirements (FR and NFR) identified • Assign a priority value for FR • Assign a priority value for NFR • Pairwise comparison for FR and NFR • Requirement prioritised	Time-consuming for large numbers of requirements
Interactive Genetic Algorithm (IGA) [51] Technical base: EVOLUTIONARY ALGORITHM	The requirement prioritisation technique is a pairwise comparison method that exploits a genetic algorithm.	• Define a set of requirements • Input orders or priorities • Initialize the population of individuals with a set of totally ordered requirements • Set a few important parameters of the algorithm • Execute IGA • Determine the fitness measure (disagreement) to be used during the next selection of the best individuals	The proposed algorithm has not been tested.
Architecture-Driven Quality Requirements Prioritization [53] Technical base: DECISION MAKING	A method employs an automated design space exploration technique based on quantitative evaluation of quality at- tributes of software architecture models.	• Modelling • Architecture model • Select applicable quality degrees of freedom • Operationalize quality requirements • Model effect • Automated Exploration; Automated design space exploration • Analysis • Conflict and dependencies detection • Manual analysis with decision support	This study does not use a technique to handle dependencies specifically. The proposed method has not been tested in industrial setting.

Technique	Description	Process	Limitation
Liquid-Democracy-based Requirements Prioritization [54] Technical base: DECISION MAKING	In terms of liquid democracy, stakeholders can either evaluate the interest dimensions directly or delegate their vote for a specific interest dimension (or requirement) to a stakeholder who is more qualified to evaluate this dimension/ requirement.	• Group members had to provide a single rating (1-5 stars) for every requirement. • a group-based multi-attribute utility (MAUT) based approach was used to determine a prioritization • The members were asked to comment on issues for every requirement. Every requirement was discussed individually by the group	It supports a small number of features and The proposed method has not been tested in industrial setting.
Supervised Classification Techniques [55] Technical base: MACHINE LEARNING	Introduce an intelligent system in order to tackle the open issues regarding dependencies between requirements by using supervised learning techniques based on text-mining. .	• First, showed 30 different requirements regarding a sports watch to each participant. • the second step, the set of randomly ordered requirements was shown to participants. The participants were asked to manually find all correct dependencies of type requires between two requirements based on the shown title and description. • After collecting all the dependencies, Natural Language Processing techniques have been exploited to support the automated detection of dependencies	Limited to support type of dependecy: requires.
Dependency-aware software release planning (DA-SRP) [57] Technical base: GRAPH	Dependency aware software release planning (DA-SRP) maximizes the overall value of an optimal subset of features while considering the influences of value-related dependencies extracted from user preferences.	• The process starts with identification of value-related dependencies from collected user preferences. • Identified value-related dependencies will be modeled using the algebraic structure of fuzzy graphs • the resulting model is referred to as the Feature Dependency Graph (FDG) of the system • Finally, perform dependency-aware release planning to find an optimal configuration of the features using the proposed integer programming model.	Limited to support type of dependecy: value-related dependencies
MOSAs [58] Technical base: EVOLUTIONARY ALGORITHM Integrating Active Learning with Ontology-Based Retrieval [60] Technical base: MACHINE LEARNING	MOSAs were designed for generic optimization problems, and therefore they do not make any domain/problem-specific assumptions when applied AL is a form of ML in which a learning algorithm interactively queries an oracle (typically a human expert) to obtain the desired label for new data points. It has been effective in reducing human efforts in the data analysis process	• Collect requirements • Set specific cost overrun probability distribution • Apply URP • Requirements permutation • Requirements Review • Requirements Dependency Extraction by Active Learning (RD-AL) • Requirements Dependency Extraction by Ontologies • Natural Language Pre-processor Pipeline	Handle the attribute values of requirements are static The ontology depend on context engineering

Table 7 maps the techniques across the dependency types. It can be seen that most techniques emphasize on Internal dependency, particularly on Functional. Among the three Functional variations, Combination is the most explored. Across the dependency types, DRank (ML) is the most applied technique. However, DRank is used so far to address Internal-Functional dependency only.

Table 7. Mapping of Techniques and Types of Requirements Dependency

Technical-base	Technique	INTERNAL						EXTERNAL								
		Functional				*Structural*		*Time-Related*		*Value-Related*		*Human Resources*		*Business Process*		
		Combination														
		A*	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Decision Making	AHP	■	■													
	Hierarchical Dependencies							■								
	Collaborative requirement prioritisation method	■	■				■									
	Utility-based prioritisation				■											
	Majority Voting Goal-Based (MVGB)				■											
	Architecture-Driven Quality Requirements Prioritization															
	Liquid-Democracy-based	■	■													
Evolutionary Algorithm	Least-Squares-Based Random				■											
	Early mutation testing				■											
	Hybrid Enriched Genetic Revamped Integer Linear Programming								■							
	Multi-objective evolutionary algorithms (MOEAs)															
	Interactive Genetic Algorithm (IGA)															
	MOSAs	■								■	■					
Fuzzy Logic	Hierarchical Fuzzy Inference System (HFIS)	■	■		■											
	Tensor and Fuzzy Graphs									■						
	Fuzzy Inference System (FIS)															
Graph	Software Features Prioritisation			■	■			■								
	Dependency-Aware Software Release Planning (DA-SRP)									■	■					
Graph & Math	Graphs and Integer Programming	■	■													
Machine Learning	Collaborative requirement prioritization approach (CDBR)					■	■									
	DRank	■	■	■	■											
	Interactive Next Release Problem (iNRP)			■	■											
	Integrating Active Learning with Ontology-Based Retrieval	■	■													
	Supervised Classification Techniques	■	■													
Matrix	Requirements Change Analysis				■			■								
	Hidden Structure Method					■										
	Traceability Metrics															
Matrix & Graph	Component-Based Software Development (CBSD)															
NLP	Satisfiability Modulo Theories (SMT)															
	SNIPR															
Technical Scale	Enhancing the Process of Requirements Prioritization in Agile Software Development	■	■		■											

[a] A: Complete, B: Limited, C: Exclusion, D: Implication, E: Direct, F: Indirect, G: Refines, H: Time-Based, I: Cost-Based, J: Revenue-Based, K: Dependencies due to downstream activities, L: Team-based dependencies, M: Inter-domain dependencies, N: Intra-domain dependencies, O: Dependencies among user stories.

The least explored techniques are AHP Modified (Decision Making), Fuzzy Graph and Tensor Algebra (Fuzzy Logic), Hidden Structure (Matrix), Interactive GA (EA) and Hybrid Enriched Genetic Revamped Integer Linear Programming (EA). The least investigated Internal dependency is Structural-Refines. In comparison to Internal dependency, External dependency receives lesser attention. In fact, only Time-related and Value-related dependencies are being addressed involving merely three techniques, namely Fuzzy Graph and Tensor Algebra (Fuzzy Logic), Genetic Algorithm (EA) and Hybrid Enriched Genetic Revamped Integer Linear Programming (EA).

Most of the techniques proposed by the authors are still at the research stage and have not been applied to solve real-world industry problems. One technique that has been applied in real-world cases is DA-SRP [57]. This technique is used in the development of industrial software called PMS-II. The case study involves 23 features considering user preferences within a certain budget. The preference matrix for PMS-II is constructed based on user preferences. The calculation of dependency strength and quality related to values is then performed using fuzzy membership functions. This technique results in optimal feature selection. Another technique is Integrating Active Learning with Ontology-Based Retrieval [60]. This technique is applied to two industrial datasets, namely Siemens Austria and Blackline Safety Corp Canada. Both companies have collected software requirements and manually determined RD. The application of the proposed technique in the research reduces efforts with good accuracy, achieving 86% accuracy in the second company.

Based on these findings, several preliminary interpretations can be made. One possible explanation on why most techniques address Internal dependency is because such dependency is definite and structured. Thus, it is more straightforward to tackle. On the other hand, External dependency involves vague and diverse elements that rely heavily on the nature of project. The elements in fact vary across projects, which are not so apparent to determine. This also helps to explain why among External dependency types, only Cost-based and Time-based are addressed by the techniques. This is due to the fact that cost and time are the most objective variables in projects. Another challenge in handling external dependencies is the conditions that are beyond the control of system developers. External dependencies can be addressed by involving stakeholders as business owners or parties related to the system requirements being developed. Considerations from these stakeholders can be used as an important factor in determining the priority sequence.

5. Conclusion

This study has provided an understanding of requirements dependency in RP in terms of its impacts, types and techniques based on a review made on thirty selected articles. The results show that requirements dependency has significant impacts on RP. Ignoring requirements dependency during RP could delay product release and increase project cost as well as project risk. The different types of requirements dependency have also been identified. There are at least 14 types, which can be clustered into two categories: Internal and External. Each type has different characteristics and thus requires different techniques. There are 28 techniques that are capable of handling requirements dependency problems in RP. These techniques are derived from various technical bases, including Fuzzy Logic, Decision Making, Evolutionary Algorithm, Matrix, Machine Learning, Graph, and Neuro-Linguistic Programming.

Some limitations and gaps are observed in the reviewed articles, which require further research. Most techniques focus on Internal dependency, rather than External. In fact, Functional is more investigated than Structural in Internal dependency. With regards to practicality, most techniques to date are still being tested in laboratory settings with small data sets and covering limited types of dependency. Their scalability and efficiency in handling large-scale requirements are thus arguable. Future studies should be able to apply RP techniques by considering dependency factors with various types in large-scale software development with a set of requirements. As RP plays an important role in ensuring the success of a software project, effective and yet practical solutions are necessary. In prioritizing requirements that involve multiple factors and a large number of requirements, combining Multicriteria Decision Making techniques with Machine Learning can be beneficial. However, it requires adjustments based on business value. In this study, the QAC process was performed manually without the use of tools. Therefore, future reviews can utilize tools such as the Cochrane Risk of Bias tool to obtain stronger assessment results.

Acknowledgment

The authors would like to thank Universiti Kebangsaan Malaysia (PP-FTSM-2020) and Universitas Ahmad Dahlan for supporting this research.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] S. Fahmy and J. H. Yahaya, "The Role of Human in Software Configuration Management," in *The 2018 7th International Conference on Software and Computer Applications*, 2018, pp. 56–60, doi: 10.1145/3185089.3185117.

[2] F. Hujainah, R. Binti, A. B. U. Bakar, and M. A. Abdulgabber, "Investigation of Requirements Interdependencies in Existing Techniques of Requirements Prioritization," *Hrčak*, vol. 26, no. 4, pp. 1186–1190, 2019, doi: 10.17559/TV-20171129125407.

[3] N. Garg, "Recent Advancements in Requirement Elicitation and Prioritization Techniques," in *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India: IEEE, 2015, pp. 237–240. doi: 10.1109/ICACEA.2015.7164702.

[4] M. Soumya Krishnan, "RFP based Requirement Prioritization - A One-Step Solution," *Mater. Today Proc.*, vol. 5, no. 1, pp. 642–649, 2018, doi: 10.1016/j.matpr.2017.11.128.

[5] P. Bajaj and V. Arora, "Multi-Person Decision-Making for Requirements Prioritization using Fuzzy AHP," *ACM SIGSOFT Software Engineering Notes*, vol. 38, no. 5, pp. 1–6, 2013. doi: 10.1145/2507288.2507302.

[6] R. H. Al-Ta'ani and R. Razali, "A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, p. 846, 2016, doi: 10.18517/ijaseit.6.6.1375.

[7] M. Alrashoud, M. Al-hammadi, A. Ghoneim, E. Hazza, F. Alqahtani, and A. Abhari, "Cognitive and Hierarchical Fuzzy Inference System for Generating Next Release Planning in SaaS Applications," *IEEE Access*, vol. 7, pp. 102966–102974, 2019, doi: 10.1109/ACCESS.2019.2929214.

[8] M. Masood, F. Azam, M. W. Anwar, and A. Amjad, "Defining Meta-Model for Value-Oriented Requirement Prioritization Technique," in *Proceedings of the 2019 7th International Conference on Computer and Communications Management*, Bangkok, Thailand: ACM New York, NY, USA, 2019, pp. 72–77. doi: https://doi.org/10.1145/3348445.3352739.

[9] R. K. Chopra, V. Gupta, and D. S. Chauhan, "Experimentation on accuracy of non functional requirement prioritization projects ☆," *Perspect. Sci.*, vol. 8, pp. 79–82, 2016, doi: 10.1016/j.pisc.2016.04.001.

[10] M. A. Rahman, R. Razali, and D. Singh, "A Risk Model of Requirements Change Impact Analysis," *J. Softw.*, vol. 9, no. 1, pp. 76–81, 2014, doi: 10.4304/jsw.9.1.76-81.

[11] N. C. Pa and A. M. Zain, "A Survey of Communication Content in Software Requirements Elicitation involving Customer and Developer," *J. Softw. Syst. Dev.*, vol. 2011, pp. 1-12, 2011, doi: 10.5171/2011.742200.

[12] A. Gupta and C. Gupta, "Towards Dependency Based Collaborative Method for Requirement Prioritization," *2018 11th Int. Conf. Contemp. Comput. IC3 2018*, pp. 1–3, 2018, doi: 10.1109/IC3.2018.8530542.

[13] H. Sheemar, “Enhancing User-Stories Prioritization Process in Agile Environment,” in *2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI)*, Greater Noida, India: IEEE, pp. 1-6, 2017. doi: 10.1109/ICICCIS.2017.8660760.

[14] F. F. Ismail, R. Razali, and Z. Mansor, “Considerations for Cost Estimation of Software Testing Outsourcing Projects,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 1, pp. 142–152, 2019. doi: 10.18517/ijaseit.9.1.6382.

[15] A. Sureka, “Requirements Prioritization and Next-Release Problem under Non-Additive Value Conditions,” in *2014 23rd Australian Software Engineering Conference*, Milsons Point, NSW, Australia: IEEE, 2014, pp. 120–123. doi: 10.1109/ASWEC.2014.12.

[16] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. R. Mahrin, “A systematic literature review of software requirements prioritization research,” *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014, doi: 10.1016/j.infsof.2014.02.001.

[17] N. Kukreja, S. S. Payyavula, B. Boehm, and S. Padmanabhuni, “Value-based requirements prioritization: Usage experiences,” *Procedia Comput. Sci.*, vol. 16, pp. 806–813, 2013, doi: 10.1016/j.procs.2013.01.084.

[18] A. Mustapha and N. Ibrahim, “Minimizing Inter-Dependency Issues Of Requirements In Parallel Developing Software Projects With Ahp,” *COMPUSOFT, An Int. J. Adv. Comput. Technol.*, vol. 8, no. Viii, pp. 3317–3323, 2019. Available at: https://search.proquest.com/openview/be41202e4771ac573941bf7633203a2c/1?pq-origsite=gscholar&cbl=2032622.

[19] A. Martakis, “Handling Requirements Dependencies in Agile Projects : A Focus Group with Agile Software Development Practitioners,” in *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, Paris, France: IEEE, 2013, pp. 1–11. doi: 10.1109/RCIS.2013.6577679.

[20] T. S. G. International, “Chaos Report 2015,” pp. 1-13, 2015. Available at: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

[21] F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, and K. Z. Zamli, “Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges,” *IEEE Access*, vol. 6, pp. 71497–71523, 2018, doi: 10.1109/ACCESS.2018.2881755.

[22] T. Amelia and R. B. Mohamed, “Review on Cost-Value Approach for Requirements Prioritization Techniques,” *Proc. - 2018 5th Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2018*, pp. 310–314, 2018, doi: 10.1109/ICITACEE.2018.8576908.

[23] F. Sher, D. N. A. Jawawi, R. Mohamad, and M. I. Babar, “Requirements prioritization techniques and different aspects for prioritization a systematic literature review protocol,” *2014 8th Malaysian Softw. Eng. Conf. MySEC 2014*, no. December, pp. 31–36, 2014, doi: 10.1109/MySec.2014.6985985.

[24] M. Sufian, Z. Khan, S. Rehman, and W. H. Butt, “A Systematic Literature Review : Software Requirements Prioritization Techniques,” in *2018 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, Pakistan: IEEE Computer Society, 2018, pp. 35–40. doi: 10.1109/FIT.2018.00014.

[25] A. M. Pitangueira, R. S. P. Maciel, M. De Oliveira Barros, and A. S. Andrade, “A systematic review of software requirements selection and prioritization using SBSE approaches,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8084 LNCS, pp. 188–208, 2013, doi: 10.1007/978-3-642-39742-4_15.

[26] R. H. Al-Ta'ani and R. Razali, “A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, p. 846, 2016, doi: 10.18517/ijaseit.6.6.1375.

[27] B. Kitchenham *et al.*, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” Durham, UK, pp. 1-65, 2007. Available at: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[28] N. Misaghian, H. Motameni, and M. Rabbani, “Prioritizing interdependent software requirements using tensor and fuzzy graphs,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 4, pp. 2697–2717, 2019, doi: 10.3906/elk-1806-179.

[29] A. Gupta and C. Gupta, "CDBR: A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 2, pp. 421-432, 2018, doi: 10.1016/j.jksuci.2018.10.004.

[30] H. Ahuja and U. Batra, "Performance Enhancement in Requirement Prioritization by Using Least-Squares-Based Random Genetic Algorithm," in *Innovations in Computational Intelligence*, Singapore: Springer, Singapore,vol. 713, pp. 251–263, 2018. doi: 10.1007/978-981-10-4555-4_17

[31] L. Alawneh, "Requirements Prioritization Using Hierarchical Dependencies," in *Advances in Intelligent Systems and Computing*, AISC, Volu.Cham: Springer, Cham, vol. 558, pp. 459–464, 2018. doi: 10.1007/978-3-319-54978-1.

[32] N. Condori-Fernandez, M. F. Granda, and T. E. J. Vos, "Towards a functional requirements prioritization with early mutation testing," *2018 IEEE/ACM 5th Int. Work. Requir. Eng. Test.*, pp. 21–24, 2018, doi: 10.1145/3195538.3195539.

[33] A. Felfernig and M. Stettinger, "Towards Utility-based Prioritization of Requirements in Open Source Environments," in *2018 IEEE 26th International Requirements Engineering Conference*, IEEE, Ed., Banff, Canada, 2018, pp. 406–411. doi: 10.1109/RE.2018.00-17.

[34] S. Jayatilleke, R. Lai, and K. Reed, "A method of requirements change analysis," *Requir. Eng.*, vol. 23, no. 4, pp. 493-508, 2017, doi: 10.1007/s00766-017-0277-7.

[35] F. Shao, R. Peng, H. Lai, and B. Wang, "DRank: A semi-automated requirements prioritization method based on preferences and dependencies," *J. Syst. Softw.*, vol. 126, pp. 141–156, 2017, doi: 10.1016/j.jss.2016.09.043.

[36] M. Alkandari and A. Al-shammeri, "Enhancing the Process of Requirements Prioritization in Agile Software Development - A Proposed Model," *J. Softw.*, vol. 12, no. 6, pp. 439–453, 2017, doi: 10.17706/jsw.12.6.439-453.

[37] R. M. Liaqat, M. A. Ahmed, F. Azam, and B. Mehboob, "A Majority Voting Goal Based technique for Requirement Prioritization," *2016 22nd Int. Conf. Autom. Comput. ICAC 2016 Tackling New Challenges Autom. Comput.*, pp. 435–439, 2016, doi: 10.1109/IConAC.2016.7604958.

[38] Z. Peng, J. W. B, K. He, and H. Li, "An Approach for Prioritizing Software Features Based on Node Centrality in Probability," in *Computer Science*, in vol 9679, vol. 1. Cham: Springer, Cham, 2016, pp. 106–121. doi: 10.1007/978-3-319-35122-3.

[39] A. Allex, A. Matheus, I. Yeltsin, A. Dantas, and J. Souza, "An Architecture based on interactive optimization and machine learning applied to the next release problem," *Autom. Softw. Eng.*, vol. 24, no. 3, pp. 623–671, 2016, doi: 10.1007/s10515-016-0200-3.

[40] D. Mougouei, "Factoring Requirement Dependencies in Software Requirement Selection using Graphs and Integer Programming," in *2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Singapore: IEEE International Conference, 2016, pp. 884–887. doi: 10.1145/2970276.2975936

[41] R. Lagerström, M. Addibpour, and F. Heiser, "Product Feature Prioritization Using the Hidden Structure Method: A Practical Case at Ericsson," in *2016 Proceedings of PICMET '16: Technology Management for Social Innovation*, Honolulu, Hawaii, USA: PICMET, 2016, pp. 2308–2315. doi: 10.1109/PICMET.2016.7806519.

[42] S. Valsala and A. R. Nair, "Requirement Prioritization and Scheduling in Software Release Planning Using Hybrid Enriched Genetic Revamped Integer Linear Programming Model Karpagam University , Coimbatore , Scientist Department , Bangalore , India," *Res. J. Appl. Sci. Eng. Technol.*, vol. 12, no. 3, pp. 347–354, 2016, doi: 10.19026/rjaset.12.2342.

[43] R. Alzyoudi, K. Almakadmeh, and H. Natoureah, "A Probability Algorithm for Requirement Selection In Component-Based Software Development," *IPAC '15 Proc. Int. Conf. Intell. Inf. Process. Secur. Adv. Commun.*, pp. 1–6. doi: 10.1145/2816839.2816871.

[44] P. R. B and M. Patrick, "Estimating the Implementation Risk of Requirements in Agile Software Development Projects with Traceability Metrics," in *International Working Conference on Requirements*

Engineering: Foundation for Software Quality, Switzerland: Springer International Publishing, 2015, pp. 81–97. doi: 10.1007/978-3-319-16101-3.

[45] M. Alrashoud and A. Abhari, "Intelligent Automation & Soft Computing Perception-Based Software Release Planning," *Intell. Autom. Soft Comput.*, vol. 21, no. 2, pp. 175–195, 2015, doi: 10.1080/10798587.2014.960229.

[46] J. Cox, E. Bouwers, M. Van Eekelen, and J. Visser, "Measuring Dependency Freshness in Software Systems," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, Italy: IEEE, 2015, pp. 109–118. doi: 10.1109/ICSE.2015.140.

[47] A. Sureka, "Requirements Prioritization and Next-Release Problem under Non-Additive Value Conditions," in *2014 23rd Australasian Software Engineering Conference*, Sydney: IEEE, 2014, pp. 120–123. doi: 10.1109/ASWEC.2014.12.

[48] Z. Tong, Q. Zhuang, Q. Guo, and P. Ma, "Research on Technologies of Software Requirements Prioritization," in *International Conference on Trustworthy Computing and Services ISCTCS 2013*, Beijing, China: Springer, Berlin, Heidelberg, 2014, pp. 9–21. doi: 10.1007/978-3-662-43908-1.

[49] J. Mczara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers — a controlled experiment," *Empir. Softw. Eng.*, vol. 20, no. 6, pp. 1721–1761, 2014, doi: 10.1007/s10664-014-9334-8.

[50] F. Fellir, K. Nafil, and Rajaa Touahni, "System requirements prioritization based on AHP," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, Tetouan, Morocco: IEEE, 2014, pp. 163–167. doi: 10.1109/CIST.2014.7016612.

[51] P. Tonella, A. Susi, and F. Palma, "Interactive requirements prioritization using a genetic algorithm," *Inf. Softw. Technol.*, vol. 55, no. 1, pp. 173–187, 2013, doi: 10.1016/j.infsof.2012.07.003.

[52] M. Daneva *et al.*, "Agile requirements prioritization in large-scale outsourced system projects : An empirical study," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1333–1353, 2013, doi: 10.1016/j.jss.2012.12.046.

[53] A. Koziolek, "Architecture-Driven Quality Requirements Prioritization," in *2012 First IEEE International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks)*, Chicago, Illinois, USA: IEEE, 2012, pp. 15–19. doi: 10.1109/TwinPeaks.2012.6344554.

[54] R. Samer, M. Stettinger, and A. Felfernig, "Group Recommender User Interfaces for Improving Requirements Prioritization," *UMAP 2020 - Proc. 28th ACM Conf. User Model. Adapt. Pers.*, pp. 221–229, 2020, doi: 10.1145/3340631.3394851.

[55] M. Atas, R. Samer, and A. Felfernig, "Automated Identification of Type-Specific Dependencies between Requirements," *Proc. - 2018 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2018*, pp. 688–695, 2019, doi: 10.1109/WI.2018.00-10.

[56] H. Saeeda, J. Dong, Y. Wang, and M. A. Abid, "A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project," *J. Softw. Evol. Process*, vol. 32, no. 7, pp. 1–24, 2020, doi: 10.1002/smr.2247.

[57] D. Mougouei and D. M. W. Powers, "Dependency-aware software release planning through mining user preferences," *Soft Comput.*, vol. 24, no. 15, pp. 11673–11693, 2020, doi: 10.1007/s00500-019-04630-y.

[58] H. Zhang, M. Zhang, T. Yue, S. Ali, and Y. Li, "Uncertainty-wise Requirements Prioritization with Search," *ACM Trans. Softw. Eng. Methodol.*, vol. 30, no. 1, pp. 1-54, 2021, doi: 10.1145/3408301.

[59] F. Hujainah, R. Binti Abu Bakar, A. B. Nasser, B. Al-haimi, and K. Z. Zamli, "SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects," *Inf. Softw. Technol.*, vol. 131, no. November 2020, p. 106501, 2021, doi: 10.1016/j.infsof.2020.106501.

[60] G. Deshpande *et al.*, "Requirements Dependency Extraction by Integrating Active Learning with Ontology-Based Retrieval," *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2020-Augus, pp. 78–89, 2020, doi: 10.1109/RE48521.2020.00020.

[61] A. Ali, Y. Hafeez, S. Hussain, and S. Yang, "Role of Requirement Prioritization Technique to Improve the Quality of Highly-Configurable Systems," *IEEE Access*, vol. 8, pp. 27549–27573, 2020, doi: 10.1109/ACCESS.2020.2971382.

[62] A. Vescan and C. Serban, "Towards a new Test Case Prioritization Approach based on Fuzzy Clustering Analysis," *Proc. - 2020 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2020*, pp. 786–788, 2020, doi: 10.1109/ICSME46990.2020.00091.

[63] M. Sadiq and V. S. Devi, "A rough-set based approach for the prioritization of software requirements," *Int. J. Inf. Technol.*, vol. 14, no. 1, pp. 447–457, 2022, doi: 10.1007/s41870-021-00749-0.

[64] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, "An industrial survey of requirements interdependencies in software product release planning," *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 84–91, 2001, doi: 10.1109/ISRE.2001.948547.

[65] C. Li, M. van den Akker, S. Brinkkemper, and G. Diepen, "An integrated approach for requirement selection and scheduling in software release planning," *Requir. Eng.*, vol. 15, no. 4, pp. 375–396, May 2010, doi: 10.1007/S00766-010-0104-X.

[66] Å. G. Dahlstedt and A. Persson, "Requirements Interdependencies : State of the Art and Future Challenges," in *Engineering and Managing Software Requirements*, Springer, pp. 95-116, 2005. doi: 10.1007/3-540-28244-0_5

[67] G. Deshpande, "SReYantra: Automated software requirement inter-dependencies elicitation, analysis and learning," *Proc. - 2019 IEEE/ACM 41st Int. Conf. Softw. Eng. Companion, ICSE-Companion 2019*, pp. 186–187, May 2019, doi: 10.1109/ICSE-COMPANION.2019.00076.

[68] A. Felfernig *et al.*, "Towards utility-based prioritization of requirements in open source environments," *Proc. - 2018 IEEE 26th Int. Requir. Eng. Conf. RE 2018*, pp. 406–411, 2018, doi: 10.1109/RE.2018.00-17.

[69] T. L. Saaty, "Decision Making – The Analytic Hierarchy And Network Processes (AHP / ANP)," *J. Syst. Sci. Syst. Eng.*, vol. 13, no. 1, pp. 1–35, 2004. doi: 10.1007/s11518-006-0151-5.

[70] J. McZara, S. Sarkani, T. Holzer, and T. Eveleigh, "Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment," *Empir. Softw. Eng.*, vol. 20, no. 6, pp. 1721–1761, 2015, doi: 10.1007/s10664-014-9334-8.

[71] R. Lagerström, M. Addibpour, and F. Heiser, "Product feature prioritization using the Hidden Structure method: A practical case at Ericsson," *PICMET 2016 - Portl. Int. Conf. Manag. Eng. Technol. Technol. Manag. Soc. Innov. Proc.*, pp. 2308–2315, 2017, doi: 10.1109/PICMET.2016.7806519.

[72] N. Condori-Fernandez and P. Lago, "Characterizing the contribution of quality requirements to software sustainability," *J. Syst. Softw.*, vol. 137, pp. 289–305, 2018, doi: 10.1016/j.jss.2017.12.005.