Journal on Interactive Systems, 2026, 17:1
ISSN: 2763-7719 • doi: 10.5753/jis.2026.5371

Research Paper

Requirement Prioritization in Software Engineering: a Systematic Literature Review Update

Renato Cesar Ais [Federal University of Technology of Paraná | *renatoais@alunos.utfpr.edu.br*]
Érica Ferreira De Souza [Federal University of Technology of Paraná | *ericasouza@utfpr.edu.br*]
Alinne C. Correa Souza [Federal University of Technology of Paraná | *alinnesouza@utfpr.edu.br*]

Federal University of Technology of Paraná, Av. Alberto Carazzai, 1640 - Centro, Cornélio Procópio - PR, 86300-000, Brazil.

Abstract. *Context:* Software requirements prioritization is the classification and ordering of requirements given their priority. This ordering can be done using prioritization techniques. Knowing the main prioritization techniques is essential for advancing research in Requirement Engineering. A Systematic Literature Review (SLR) published in 2021 presented evidence on requirements prioritization techniques, their limitations, a list of identified techniques, and the criteria used in the prioritization process. An assessment of this SLR revealed the need for an update. *Purpose:* This study aims to update the previous SLR and contribute to the current state of research on this topic. *Method:* First, we conducted a tertiary study to identify other reviews addressing the same topic. Then, we applied the 3PDF framework to evaluate the feasibility of SLR updating, and finally, we updated the previous SLR. In this update, we consider new research questions and analyses of the extracted data. This update incorporates new research questions and additional analyses of the extracted data. *Results:* The updated SLR identified a total of 45 relevant studies between 2021 and June 2025, shedding new light on the evolution of requirements prioritization. We identified 32 distinct requirements prioritization techniques, with 23 of these being novel, adding a fresh perspective to the area. *Conclusion:* This updated SLR provides a comprehensive view of the continuous evolution in requirements prioritization. The review highlights significant advancements in the use of Machine Learning (ML) and Artificial Intelligence (AI) techniques, alongside the enduring popularity of traditional methods such as the Analytic Hierarchy Process (AHP) and MoSCoW. Understanding these trends is crucial for practitioners, as they reflect persistent challenges in decision-making, including the influence of individual preferences and domain knowledge on the prioritization process.

Keywords: Requirement Engineering, Requirements Prioritization, Systematic Literature Review, Updated

Edited by: Ingrid Monteiro | **Received:** 24 December 2024 • **Accepted:** 14 October 2025 • **Published:** 01 January 2026

1 Introduction

The design of modern systems increasingly requires careful attention to the early stages of development, as initial decisions significantly influence the overall quality, usability, and effectiveness of the resulting solutions [Monteiro and Batista, 2023; Ribeiro and Garcés, 2023]. As application domains expand and both technical and social complexities grow, it becomes imperative to adopt systematic practices for planning and specifying requirements. Such practices support stakeholders in organizing and prioritizing system functionalities, ultimately enabling the development of more robust and user-centered solutions [Ribeiro and Garcés, 2023].

The main purpose of the software industry is to meet customer expectations effectively and with quality by defining the software requirements of projects. Requirements are descriptions of activities that software must perform in order for software to solve a real-world problem. The process of identifying, documenting, and monitoring these requirements is the responsibility of Requirement Engineering [Bourque *et al.*, 2014].

Requirement Engineering makes the connection between the client and the software project [Pressman and Maxim, 2014], as it enables the definition of specific restrictions and objectives for each project, allowing interested parties to make informed choices throughout the process. This ensures the management and documentation of requirements throughout the entire software development life cycle, from functionality implementation to change management. The success or failure of the project is directly influenced by the effective application of Requirement Engineering techniques [Hussain and Mkpojiogu, 2016].

Rapid changes in business rules, technology evolution, and project size have challenged Requirement Engineering on regularly, often making requirements outdated before the project ends. This makes it necessary to more effectively manage software requirements in order to meet constant changes during the software development project [Cao and Ramesh, 2008]. When the project is divided into smaller parts, requirements management can become more effective. This is the principle adopted by agile methodologies, which provides for the division of work into iterations with short durations. With the adoption of agile methods, the requirements can be known at the beginning, but they will be detailed as they are prioritized for allocation in a given release [Reinehr, 2020].

The requirements prioritization stage in Requirement Engineering is responsible for carrying out negotiations between interested parties, evaluating costs and risks, and resolving possible conflicts of interest, with the aim of achieving the satisfaction of all parties involved in the project [Pressman and Maxim, 2014]. When prioritizing requirements, those involved in the project must reach a consensus to resolve the order in which the requirements will be implemented based on the factors established in each project [Sommerville, 2011]. A prioritizing session may consist of three main stages [Karlsson *et al.*, 1998]: (i) *Preparation* - the requirements are structured according to the prioritizing techniques to be used; (ii) *Execution* - the decision-makers prioritize the requirements based on the information provided in the previous stage; and (iii) *Presentation* - the results of the execution are presented to those involved.

The prioritization of requirements will depend on several factors, such as: the potential value added to the business by the requirement; dependencies between requirements; analysis of implicit requirements that may be overlooked; experience with technology by the project team; experience in the application domain on the part of the team; relationships with other systems (hardware or software); and implementation demands to meet legal or regulatory requirements [Reinehr, 2020; Gerogiannis *et al.*, 2022]. To perform requirements prioritization, techniques that support the prioritization process can be used, regardless of the factors involved.

Various requirements prioritization techniques have been developed to support software development teams, from conventional techniques using human iteration to techniques that use computational power. Some best-known techniques in the literature are: *Analytic Hierarchy Process (AHP)* [Perini *et al.*, 2007], *cost–value* [Karlsson and Ryan, 1997], *MoSCoW* [Miranda, 2022], *Grey Wolf Optimization* [Masadeh *et al.*, 2018] and *K-means Clustering* [Achimugu *et al.*, 2014]. Although there are techniques that support the prioritization of software requirements, in practice, there are still software development companies that carry out the selection process informally because they do not know how to assign priority to requirements, which can generate poor quality software products.

In Rashdan [2021], a Systematic Literature Review (SLR) was conducted to summarize the current trends in software requirements prioritization techniques, their limitations, and the processes involved. From an analysis of this SLR, we noted that new evidence could be considered, as well as new analyses. Therefore, we decided to use the decision framework to assess systematic reviews for updating, called the third-party decision framework (3PDF) [Garner *et al.*, 2016], to confirm the need for an update to Rashdan's SLR [Rashdan, 2021]. Once the need for SLR maintenance was confirmed, we upgraded Rashdan's SLR. Furthermore, we conducted an extension considering new research questions.

The remainder of this paper is structured as follows. Section 2 presents the related work. Sections 3 argue on the research method and the selection process applied to perform the SLR update. Analysis of the obtained results according to the research questions is reported in Section 4. Research opportunities are discussed in Section 5. Section 6 discusses the threats to validity. Conclusion and future research are presented in Section 7.

2 Related Works

In this study, we present an SLR update that identifies and classifies all research related to requirements prioritization techniques, their limitations, and the steps involved in the requirements prioritization process. Before conducting the SLR, we carried out a tertiary study to find other reviews that addressed the same topic. Tertiary studies are reviews that focus exclusively on secondary studies (SLRs or Mapping Studies), that is, reviews of other reviews [Keele *et al.*, 2007]. In the tertiary study conducted, the following search string was used:

("Requirements prioritization") AND ("categories" OR "taxonomies" OR "classifications"' OR "techniques" OR "activities" OR "processes" OR "limitations" OR "shortcomings" OR "practice" OR "methods" OR "practices" OR "significance") AND ("systematic literature review" OR "systematic review" OR "systematic mapping" OR "mapping study" OR "systematic literature mapping" OR "literature review")

The search string was applied to the same electronic search databases that were considered in the present study (see Section 3.2). We also searched for related secondary studies directly within the Journal on Interactive Systems (JIS) collection in the Brazilian Computer Society's SOL repository. We considered studies published between the years 2021 to 2024. From the tertiary study, seven related secondary studies were identified (**Table 1**). Some studies identified are aimed at applying requirements prioritization techniques in specific scenarios, such as in (R1) [Krishnan *et al.*, 2023], which explores the adoption of prioritization frameworks in startup companies. Another example is the study (R2) [Anwar and Bashir, 2023] which addresses requirements prioritization techniques based on Artificial Intelligence, identifying the advantages and disadvantages of these techniques. The SLR in (R6) [Nazim *et al.*, 2022] investigates techniques that use the perspectives of those involved in the process as criteria, in order to find collaborative prioritization techniques.

Secondary studies in (R3) [Alaidaros *et al.*, 2022] and (R4) [Yaseen, 2023] are more related to our SLR. In (R3), the authors searched for requirements prioritization techniques applied in project management processes. The main study objective was to answer the following research question: *What are the approaches used to prioritize the requirements of a Project?*. An analysis was carried out on 17 studies selected published between 2016 and 2021. Some techniques identified were: SRPTackle, Planning Games and Analytical Hierarchy Process (PGAHP), Associated Network of Requirement Change (ANRC), Interactive Genetic Algorithm (IGA), Apriori Algorithm, and WhaleRank. The authors highlight the WhaleRank technique in the study as it is a more efficient technique for classifying requirements. Experiments comparing whalerank with other ranking techniques, such as CBRank and Genetic Algorithms (GA), demonstrate that the WhaleRank is more efficient in terms of accuracy and disagreement measure.

In (R4), the authors investigated requirements prioritization techniques and evaluated the use of these techniques in the prioritization process of different types and sizes of requirement sets. 60 studies were analyzed and 43 prioritization techniques identified. The techniques identified with the highest number of citations were: AHP, Binary tree, Numerical assignment and Extensive (ENA), Cost value ranking and Case-based ranking. In this SLR, the authors conducted a

Table 1. Selected studies from the tertiary study

ID	Reference	Title	Year	Source
R1	[Krishnan *et al.*, 2023]	Systematic Literature Review of Product Feature Prioritization Frameworks in Startups Building Digital Products Using PRISMA	2023	IEEE Xplore
R2	[Anwar and Bashir, 2023]	A Systematic Literature Review of AI-Based Software Requirements Prioritization Techniques	2023	IEEE Xplore
R3	[Alaidaros *et al.*, 2022]	A Review on Requirements Prioritization Approaches of Software Project Management	2022	IEEE Xplore
R4	[Yaseen, 2023]	Exploratory study of existing research on software requirements prioritization: A systematic literature review	2023	Web of Science
R5	[Amelia and Mohamed, 2023]	A Review: Requirements Prioritization Criteria Within Collaboration Perspective	2023	Web of Science
R6	[Nazim *et al.*, 2022]	Fuzzy-Based Methods for the Selection and Prioritization of Software Requirements: A Systematic Literature Review	2022	Google Scholar
R7	[Talele and Phalnikar, 2021]	Classification and Prioritization of Software Requirements using Machine Learning – A Systematic Review	2021	Google Scholar

broad analysis of the prioritization techniques identified. The techniques were classified based on the number of requirements (small, medium, and large) in a given scenario. Of the 43 techniques identified, 53% were applied to scenarios with a small number of requirements, 25% to medium-sized requirements, and only 8% to large-scale requirements. This analysis highlights a prevalence of techniques designed for smaller-scale requirements, indicating a limited number of techniques applied to scenarios with a large number of requirements.

From conducting the tertiary study, it was possible to identify some related works, mainly the SLR in (R3) and (R4). Some differences between our SLR and (R3) and (R4) are: the prioritization techniques discovered by (R3) and (R4) were identified up to 2021 and, currently, there are new techniques being worked on and which were identified in our SLR; we classify the limitations of each prioritization technique identified, not only in relation to the volume of requirements as performed in (R4), but also considering all the challenges and limitations presented in the studies in general; Unlike (R3) and (R4), we conducted an analysis to identify which assessment scales were used to validate the application of prioritization techniques; finally, we also analyzed the stages of the prioritization process for each technique, observing the application processes.

3 Systematic Literature Review Update

This study updated an SLR conducted by Rashdan [2021]. An SLR was conducted in [Rashdan, 2021] to identify requirements prioritization techniques, their limitations, taxonomy, and criteria used in the requirements prioritization process. From an analysis of this SLR, we noticed the need to provide a continuous update of the state of the art on effective practices in the area of software requirements prioritization.

3.1 Update justification

As highlighted by Rashdan [2021], the continuous evolution of requirements prioritization techniques emphasizes the importance of research in the field through the updating of previously conducted studies. Rashdan [2021] identified a growing trend in the use of AI and ML techniques for requirements prioritization. With the update of this SLR, it will be possible to verify whether this trend has remained or, possibly, intensified with the advancement of these technologies. Additionally, we aim to identify the number of new requirements prioritization techniques that researchers have developed and assess the emerging challenges they face when implementing these approaches. This process will provide a deeper analysis of the continuous evolution and the challenges faced in the field of requirement engineering, highlighting how technological innovations have impacted the practices and effectiveness of prioritization techniques.

We applied the third-party decision framework (3PDF), proposed by [Garner *et al.*, 2016], to evaluate the feasibility of updating an SLR. [Garner *et al.*, 2016] presented a decision structure with parameters that show the need to update a given secondary study that goes beyond the existence of new published evidence on the topic investigated. Researchers initially proposed the 3PDF framework for studies in the medical domain, but over time, they expanded its application to other fields, including software engineering. As a result, the framework evolved into a generic approach applicable to any SRL. Researchers do not need prior evidence of new studies or methods to apply 3PDF. Instead, they can use it proactively to identify and screen potential updates, assess feasibility, and justify the need for an SRL update. In the context of Software Engineering, Mendes *et al.* [2020] applied and recommended the use of this framework. In this framework, we divide the evaluation into three stages: (i) evaluating the relevance of the topic, (ii) identifying new methods and studies, and (iii) checking the impact of the update. Each stage presents some questions that must be answered, with valid answers being **"Yes"**, **"No"**, or **"Maybe"**.

Stage 1: We begin this stage by answering three questions to evaluate the relevance of the topic. If we answer any of them with a value other than "Yes", we must interrupt the process and avoid proceeding to the next stage, which makes it impossible to update the SLR.

Stage 2: We answer two questions to identify whether new methods and studies have emerged on the topic. To proceed to the final stage, we must answer at least one of these questions with "Yes".

Stage 3: We answer two questions to determine whether the update will impact the results and credibility of the original review. If we answer at least one question with "Yes" or "Maybe", we can proceed to prepare an update to the original review.

Applying the 3PDF framework to the original SLR [Rashdan, 2021], we have the following result, as shown in Table 2, which addresses a current research question in a field that is constantly evolving, such as requirement engineering, considering the continuous development of requirements prioritization techniques in software engineering. According to the guidelines of the 3PDF framework, indicators of good access are used, such as the number of citations, accesses, downloads, and social media shares. The SRL [Rashdan, 2021] has received a total of 2030 downloads and 750 accesses by 2023, as recorded on the DIVA portal[1]. Based on these numbers, we consider the study to have significant relevance and a good level of access.

From the evaluation of the result presented, it is evident that it is viable and essential to prepare an update of the SLR presented in [Rashdan, 2021]. The objective in updating the SLR is to expand the results and promote the continuous evolution of the topic of requirement engineering techniques.

Table 2. 3PDF framework application for updating

Step	Questions	Answers
1	**Q1** (relates to Step 1.a. in the 3PDF): Does the published SLR still address a current question?	YES
	Q2 (relates to Step 1.b. in the 3PDF): Has the SLR had good access or use?	YES
	Q3 (relates to Step 1.c. in the 3PDF): Has the SLR used valid methods and was well-conducted?	YES
2	**Q4** (relates to Step 2.a. in the 3PDF): Are there any new relevant methods?	MAYBE
	Q5 (relates to Step 2.b. in the 3PDF): Are there any new studies or information?	YES
3	**Q6** (relates to Step 3.a. in the 3PDF): Will adopting new methods change the findings, conclusions, or credibility?	YES
	Q7 (relates to Step 3.b. in the 3PDF): Will including new studies/information/data change findings, conclusions or credibility?	YES

3.2 Research method

SLR in [Rashdan, 2021] was conducted considering the period between the years 2014 and 2020. We perform the update in search of primary studies published in electronic databases for the years 2021 to June 2025. This update review involved three main phases [Kitchenham, 2012]: (i) **Planning**: refers to identifying a need for update the review, and establishing a review protocol (research questions, inclusion and exclusion criteria, sources of studies, search string); (ii) **Conducting**: searches and selects the studies, to extract and synthesize data; and (iii) **Reporting**: answer the research questions, writing up the results, and circulating them to potentially interested parties. A summary of the protocol conducted in this SLR update is presented below.

Research Questions (RQ). In addition to the RQs presented in the original SLR, we created two more RQs (RQ5 and RQ6) to extend knowledge on the research topic. The new questions aimed to identify the types and evaluation methods of research carried out during the update period. RQs are presented in **Table 3**.

Search string. The search string is the same as the original SLR. The search string explores the topic of prioritizing requirements, including terms related to categories, taxonomies, classifications, techniques, activities, processes, limitations, deficiencies, practices, and methods. The search string used is presented below:

("Requirements prioritization") AND ("categories" OR "taxonomies" OR "classifications" OR "techniques" OR "activities" OR "processes" OR "limitations" OR "shortcomings" OR "practice" OR "methods" OR "practices" OR "significance")

Sources. We applied the search string to the same six electronic databases used in the original SLR: IEEE Xplore[2], ACM Digital Library[3], Science Direct[4], Web of Science[5], Springer[6] and Google Scholar[7].

Selection criteria. The selection criteria are organized in three Inclusion Criteria (IC) and eight Exclusion Criteria (EC). All three inclusion criteria must be true for the study to continue. The inclusion criteria are:
(IC1) The study concentrates on requirements prioritization;
(IC2) The study contributes to at least one of the research questions; and
(IC3) The study was published between 2021 and June 2025.

The exclusion criteria are:
(EC1) The manuscript does not contribute to any of the research questions;
(EC2) Abstract or extended abstract without full text;
(EC3) The manuscript does not have sufficient bibliographical information, for instance, the publisher;
(EC4) Study is not written in English;
(EC5) Study is a copy or an older version of another publication already considered. In these cases, the most current version is considered;
(EC6) No access to the full study;
(EC7) Not meet the inclusion criteria (IC1, IC2 and IC3); and
(EC8) Study is not a primary study.

[1] **DIVA portal** - https://www.diva-portal.org/ - Accessed on 10 January 2025

[2] **IEEE Xplore** - https://ieeexplore.ieee.org/ Accessed on 15 july 2025
[3] **ACM Digital Library** - https://dl.acm.org/ Accessed on 15 july 2025
[4] **Science Direct** - https://www.sciencedirect.com/ Accessed on 10 July 2025
[5] **Web of Science** - https://clarivate.com/ Accessed on 15 july 2025
[6] **Springer** - https://www.springer.com/ Accessed on 05 july 2025
[7] **Google Scholar** - https://scholar.google.com/ Accessed on 10 july 2025

Table 3. Research Questions

	Questions	Rationale
RQ1	What are the existing techniques for prioritizing software requirements?	Identify prioritization techniques in scientific studies and new techniques developed.
RQ2	What are the limitations of current software requirements prioritization techniques?	Identify application limitations of each technique.
RQ3	What is the listing of prioritization scales that each methodology demonstrates?	Classification of identified techniques using qualitative or quantitative scales.
RQ4	What are the steps involved in the process during the prioritization of software requirements?	Identify the main steps of each technique in the requirements prioritization process.
RQ5	What types of research are identified?	This RQ aims to identify the types of research conducted, according to the classification proposed by Wieringa *et al*. [2006]. The classification is divided into: (i) Evaluation Research; (ii) Solution Proposal; (iii) Philosophical Paper; (iv) Opinion Paper; and (v) Experience Paper.
RQ6	What assessment types were applied in the studies?	It seeks to identify the evaluation methods carried out in the studies, as presented by Wohlin *et al*. [2012]: (i) Experiment; (ii) Case Study; and (iii) Survey.

Data Storage. To collect and manage data during the SLR execution, we used a structured spreadsheet (data extraction form) that includes: a unique identifier (ID) for each study, status throughout the selection process, a complete bibliographic reference, and detailed answers to all research questions. This spreadsheet also served as a catalog to support data synthesis. To ensure transparency and replicability, the whole spreadsheet is publicly available on Zenodo at *https://zenodo.org/records/16812333*.

Assessment. The first author conducted the review, selected the studies, extracted the data, and performed the initial synthesis of the results. Throughout the process, the second and third authors provided continuous support, supervision, and active involvement in all key decisions. To reduce potential bias and ensure the reliability of the data analysis, we held regular weekly meetings in which all authors discussed progress, clarified doubts, and validated the conducted steps. This collaborative approach helped ensure consistency and transparency across all stages of the review.

3.3 Selection Process

We conducted the study by performing a systematic search across six databases, following a selection process guided by predefined criteria. **Figure 1** illustrates the main stages of this process. We describe how each selection stage was carried out.

Stage 1: As an initial result, 520 publications were returned. In the 1st stage, we eliminated duplicates (publications that appeared in more than one source), resulting in a total of 416 studies.

Stage 2: In the 2nd stage, we applied the selection criteria over title, abstract, and keywords, resulting in 143 studies.

Stage 3: In the 3rd stage, the selection criteria were applied considering the complete text, resulting in a set of 60 studies.

Stage 4: Although the authors had already completed a full reading and applied the selection criteria in Stage 3, they conducted another detailed review of the selected studies

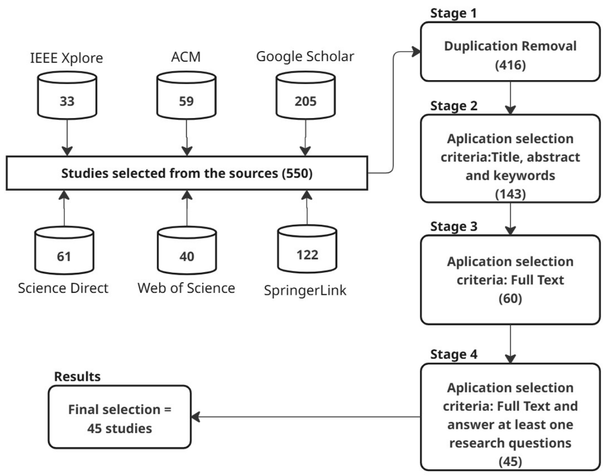

Figure 1. Search and selection SLR update process.

in Stage 4 to answer the research questions and rigorously validate the study selection. At this point, we reapplied only the exclusion criterion EC1 to ensure that each study explicitly addressed at least one of the defined research questions. Though we did not conduct a formal quality assessment, this step functioned as an informal check to ensure the methodological relevance and adequacy of the final set of studies, resulting in 45 selected papers.

As a final result, we identified 45 studies for analysis. Table 10 in Appendix A lists all selected papers along with their corresponding identifiers (IDs). Throughout this paper, we refer to each study using these IDs for clarity and consistency.

4 Results

In this section, we present the SLR finding in light of our Research Questions (RQs).

4.1 Overview of the studies

This study resulted in 45 selected studies, which are listed in Appendix A (Table 10). The studies are distributed between 2021 and June 2025. In 2021, we had the highest number of studies, with 13 (28.8%) selected. In 2022, there was a slight decline, totaling nine studies (20.0%) of the total, while in 2023, 10 studies (22.2%) were identified. In 2024, again, nine studies were identified (20%). Finally, in 2025, up to June, four studies were identified (8.8%). The mean number of studies per year was nine studies, demonstrating the continuity and evolution of research in requirements prioritization over these years.

We observed that most studies originate from Asia, with 27 studies (60.0%) of the total. Europe follows with 9 studies (20.0%), and North America with 7 studies (15.6%). South America and Oceania each contributed one study (2.2%). Among countries, Pakistan leads with seven studies, followed by India and the United States with five each. Saudi Arabia contributed four studies, while China and Spain contributed three each. These results highlight broad contributions from multiple regions to requirements prioritization techniques.

The majority of studies, 28 (62.2%), were published in journals, and 17 (37.8%) were presented at conferences. The IEEE research database stands out with the highest number of studies, totaling 17 (37.8%), followed by Springer with 11 studies (24.4%). Science Direct and Web of Science each contributed six studies (13.3%), Google Scholar provided four studies (8.9%), and the ACM database had only one study selected (2.2%).

The original SLR [Rashdan, 2021] covered a more extended period of 7 years, allowing for the inclusion of a more significant number of studies and a diverse range of publication types, while the updated SLR focused on a more recent 5-year period. Both studies identified similar challenges in applying prioritization techniques.

However, the updated SLR provided a more detailed analysis by categorizing limitations into specific groups. The original SLR addressed evaluation scales such as Ratio, Ordinal, and Nominal. In contrast, the update identified new techniques, including the Statistical Method and Hyper-Volume Metric, expanding the tools available for assessing prioritization techniques. Both studies emphasized the importance of systematically organizing and evaluating requirements to ensure an effective prioritization process.

4.2 Techniques for prioritizing software requirements (RQ1)

We identified 32 distinct techniques used in the requirements prioritization process. These techniques range from traditional methods to Machine Learning algorithms, reflecting the diversity of software development projects. Comparing the results obtained in the SLR conducted by Rashdan [2021] and the present SLR, some techniques were mentioned in both studies, as shown in Figure 2.

Analytical Hierarchy Process (AHP) stands out as the most cited and used technique in both studies. Techniques using Fuzzy Algorithms and the MoSCoW technique are also still widely studied and applied. We added all the techniques in only one study to the "Others" category (Figure 2), including: $S16$ - Goal Oriented Requirements Language [Fadel *et al.*, 2022], $S14$ - Weighted Page Rank Method [Gupta and Gupta, 2022], $S15$ - Regression Based Prioritization Technique [Malgaonkar *et al.*, 2022], $S23$ - Enhanced Analytical Hierarchy Process (E-AHP) [Mohamed *et al.*, 2022], $S10$ - FAGOSRA MS [Mohammad *et al.*, 2021], $S27$ - Criticality Factor Value (CFV) [Ahmed *et al.*, 2023], $S3$ - 360 Degree Feedback [Gerogiannis *et al.*, 2022], $S19$ - CBRank [Rojas *et al.*, 2022], $S9$ - Fuzzy C-mean [Ijaz *et al.*, 2021], $S11$ - Rough-Fuzzy DEMATEL [Zhang *et al.*, 2021], $S30$ - Branch and Bound Algorithm [del Sagrado *et al.*, 2023], $S31$ - Knap-Sack [Armah *et al.*, 2023], $S33$ - Rule-Based Automated Requirements Prioritization (RAR-P) [Izhar *et al.*, 2024], $S34$ - Improved Marine Predators Algorithm (IMPA) [Tanveer *et al.*, 2024], $S35$ - Vertical Binary Search [Brahmam *et al.*, 2024], $S36$ - Monitoring data and user feedback [Tanveer and Rana, 2024], $S43$ - Black Hole Algorithm (BHA) [Ibrahim Alfassam *et al.*, 2025], $S44$ - Active Learning and Ontological Modeling [Almoqren and Alrashoud, 2025], $S39$ - Requirement Prediction, Test-case Selection, and Prioritization (RPTSP) [Ilays *et al.*, 2024], $S45$ - Middle Mile Optimization Platform [Turkmen *et al.*, 2025], $S40$ - Social Network Analysis (SNA) [Bai *et al.*, 2024].

Among the analyzed techniques, the AHP stood out with the most citations, referenced in 10 different studies ($S1$, $S2$, $S4$, $S5$, $S7$, $S19$, $S23$, $S20$, $S24$, $S25$, $S38$, $S37$, $S42$). AHP served as a foundation for the development of new techniques. For instance, the E-AHP, presented in ($S26$), aims to overcome the limitations of the traditional AHP technique by using Machine Learning algorithms.

The Fuzzy algorithm was used in four studies ($S18$, $S2$, $S21$, $S41$) independently and in combination with other techniques. In study S2, the authors combined the Fuzzy algorithm with the AHP technique, which reduced uncertainties and subjective decisions in the requirements prioritization process, demonstrating the effectiveness of Fuzzy algorithms in this context. We also identified other techniques that incorporate Fuzzy logic, such as the Logarithmic Fuzzy Trapezoidal Approach ($S8$), Fuzzy C-means ($S9$), Intuitionistic Fuzzy Sets ($S14$), and Rough-Fuzzy DEMATEL ($S11$). Additionally, the Interactive Genetic Algorithm (IGA), mentioned in three studies ($S19$, $S28$, $S32$), employed computational algorithms to support decision-making.

In our SLR, we identified 23 new techniques, which highlight the ongoing evolution and innovation in requirements prioritization research. The Knapsack technique ($S31$) prioritized requirements by considering factors such as time and budget. Inspired by the classic "knapsack" problem, this technique selects items of varying values and weights without exceeding the knapsack's capacity. Two studies ($S21$, $S13$) applied the rough set theory technique, which helps identify essential requirements and eliminate less important ones. The CFV technique ($S27$) uses criteria defined for each project type, such as business impact, urgency, implementation complexity, and associated risks, categorizing each requirement on an ordinal scale. Additionally, the Black Hole Algorithm technique ($S43$) applies a nature inspired metaheuristic, simulating the gravitational behavior of black holes to iteratively refine and optimize the ranking of software requirements, showing promising results in large-scale and dynamic environments.

Of the 45 studies, 47% evaluated prioritization techniques in real-world scenarios, demonstrating a practical approach to their application. Meanwhile, 38% assessed the techniques using datasets from public repositories or bench-

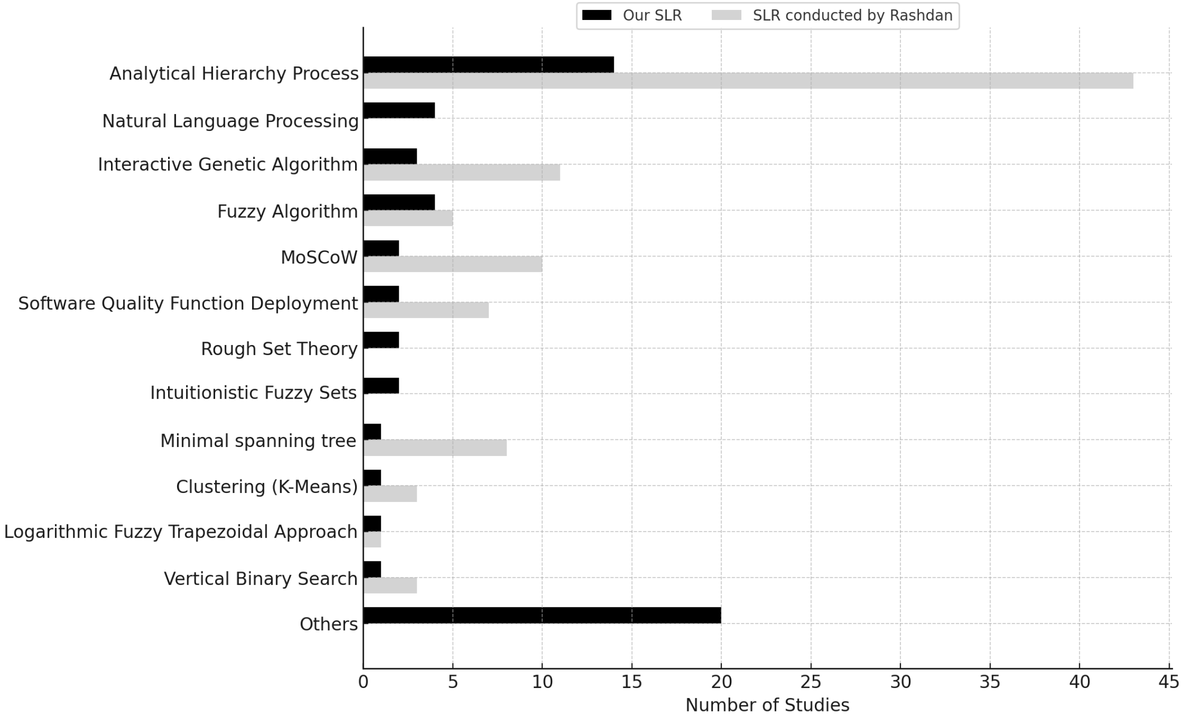

Figure 2. Techniques identified by our SLR and Rashdan [Rashdan, 2021].

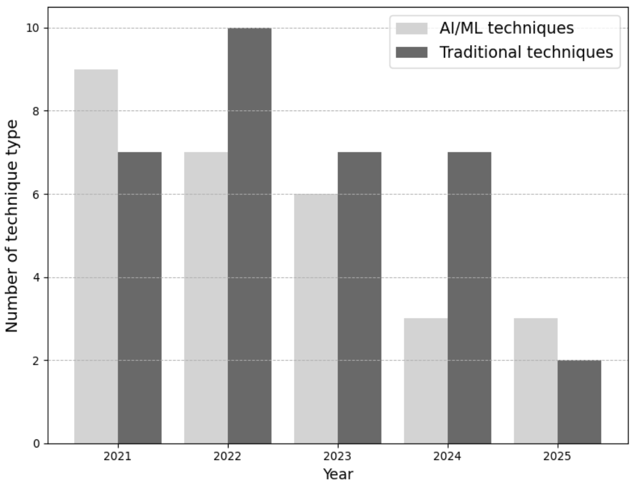

Figure 3. Distribution number of techniques based on the classification of requirements prioritization techniques by year.

mark datasets such as the Institute of Education Sciences (IES), and 15% evaluated them in theoretical scenarios within controlled environments.

Building upon Rashdan's SLR [Rashdan, 2021], the present study revealed that some techniques remain widely used, such as AHP, MoSCoW, and Fuzzy algorithms, demonstrating effectiveness and applicability in various project contexts. In addition to traditional methods, new techniques are emerging, such as the Knapsack technique and those incorporating Machine Learning algorithms like E-AHP, aimed at enhancing precision and efficiency in requirements prioritization processes.

4.2.1 Evolution of prioritization techniques: traditional vs. AI/ML-based techniques

We analyzed the evolution of requirements prioritization techniques by categorizing them into two groups: (i) traditional techniques and (ii) techniques based on Artificial Intelligence (AI) or Machine Learning (ML). Figure 3 shows their distribution over time.

Between 2021 and June 2025, the proportion of studies using AI/ML-based techniques showed variation. In 2021, 3 out of 13 studies (23%) adopted such techniques. This proportion rose to 33% in 2022 (3 out of 9), dropped significantly in 2023 to 10% (1 out of 10), and showed a slight recovery in 2025 with 25% (1 out of 4). These findings indicate that, although AI/ML techniques continue to be explored, their adoption has not followed a consistent upward trend over the years.

The AI/ML-based techniques identified were grouped into four main categories: (i) clustering methods, such as K-Means Clustering ($S6$), used to group similar requirements; (ii) natural language processing (NLP), applied in studies $S12$, $S13$, $S20$, and $S29$ to interpret and classify requirements written in natural language; (iii) metaheuristic optimization algorithms, such as the Black Hole Algorithm ($S43$), designed to improve prioritization accuracy in large-scale or dynamic contexts; and (iv) hybrid techniques, which combine Machine Learning with user feedback or domain ontologies, as seen in studies $S40$ and $S44$.

Traditional techniques such as the Analytic Hierarchy Process (AHP) and MoSCoW continue to dominate, being employed across all years analyzed. This persistent preference can be attributed to their maturity, ease of application, and alignment with collaborative decision-making processes, especially in agile environments or projects with strong stakeholder involvement.

Despite their innovative potential, AI/ML-based techniques still face significant barriers that limit their widespread use in practice. Among the main challenges reported in the literature are difficulties in scalability, handling requirement dependencies, and integrating complex algorithms into existing workflows [Talele and Phalnikar, 2021]. Additionally, the interpretability of AI/ML results remains a concern for practitioners, who often prefer transparent and explainable decision-making tools.

In summary, the findings suggest that the field is currently in a transitional phase. While AI/ML-based techniques offer promising opportunities for automation and precision in requirements prioritization, traditional methods remain more accessible, interpretable, and widely accepted in industrial settings. A balanced coexistence of both paradigms may continue in the coming years, as intelligent techniques gradually mature and become more integrated into practical software development processes.

4.2.2 Classification of requirement representations vs. prioritization techniques

The requirements prioritization techniques analyzed in this SLR employ different forms of requirement representation, which directly influence their application, automation, and comprehensibility. For categorization in this study, representations were grouped into four classes: *(i)* textual, *(ii)* formal, *(iii)* model-based, and *(iv)* hybrid. This structure aligns with the theoretical distinction proposed by [Klievtsova *et al.*, 2024], who discuss the use of textual artifacts, formal models, and hybrid representations in the context of requirement engineering. Practitioners commonly describe requirements in natural language (such as user stories). Still, they can also transform or derive them from graphical and formal models, such as Business Process Model and Notation (BPMN) diagrams or mathematical structures, using Model-to-Text (M2T) and Text-to-Model (T2M) transformations.

In this categorization, *textual* representations refer to requirements written in natural language, making them easily accessible to stakeholders, though potentially ambiguous. *Formal* representations use mathematical models or logical structures, enabling precision and automation, yet often requiring specialized knowledge. *Model-based* representations rely on visual modeling techniques, such as Unified Modeling Language (UML) or BPMN, to convey the structure, dependencies, or behaviors of the system. Finally, *hybrid* representations combine elements from two or more of these techniques, typically blending natural language with formal models in order to balance human interpretability and computational rigor.

Based on the mapping conducted (Table 4), a considerable portion of the examined techniques adopt formal representations (20 out of 32), indicating a strong inclination toward computational methods that support automation in the prioritization process. Conversely, techniques based on textual representations remain common (5 techniques), particularly in collaborative contexts where stakeholder involvement is higher and formal structure is reduced. Modeled representations were identified in two cases. Lastly, five methods employ a hybrid approach, combining textual data with formal processing in an effort to balance accessibility with analytical precision.

This result demonstrates that requirements prioritization methodologies differ not only in terms of the algorithms or structural techniques they employ, but also in how requirements are represented and interpreted throughout the decision-making process. The analysis of the studies indicates a notable predominance of techniques grounded in formal representations, particularly those incorporating optimization algorithms, fuzzy logic, heuristics, or statistical methods. These techniques stand out for their high processing capacity and automation capabilities, making them especially effective in projects involving a large number of requirements, multiple quantitative criteria, or the need for continuous reassessment, as commonly observed in continuous integration environments or agile delivery cycles. Techniques such as Fuzzy C-means, Knapsack, Rough Set Theory, and Black Hole Algorithm exemplify this group, offering computational precision and scalability with minimal human intervention.

In contrast, techniques based on textual or hybrid representations are often better suited to contexts where subjective interpretation, active stakeholder involvement, and the construction of consensus among diverse perspectives are critical factors. Techniques like MoSCoW, AHP, 360 Degree Feedback, and RAR-P deal with requirements expressed in natural language or qualitative assessments, facilitating their application in collaborative projects, agile methodologies, or early development stages, where requirements are still being refined. In such scenarios, the focus shifts away from automation and toward shared understanding, stakeholder negotiation, and the exploration of implicit preferences. Hybrid techniques such as those combining NLP techniques with formal decision models, represent a deliberate effort to balance cognitive accessibility with algorithmic support.

This mapping provides important insights for methodological decisions within a specific project, by demonstrating the feasibility of employing different forms of representation at various stages of the software lifecycle. For example, it is possible to begin the process with techniques that rely on textual representations during the requirements elicitation phase, which facilitates communication among project participants, and later transition to formal techniques during the release planning phase, where higher levels of precision, scalability, and automation in prioritization are required.

4.3 Limitations of techniques for prioritizing software requirements (RQ2)

RQ2 aims to identify the limitations of techniques for prioritizing software requirements. These limitations can range from the computational complexity required for implementing the technique to issues related to the geographical distribution of project teams and human factors, such as the need for more knowledge about business rules during the requirements classification. We classified the limitations into 8 categories according to their specific characteristics, as identified during the analysis, allowing for a detailed and systematic organization of the information as presented in the Table 5.

We observed ten techniques associated with the **Human Factors** category. These techniques include MoSCoW ($S6$, $S22$), 360 Degree Feedback ($S3$), AHP ($S1$, $S2$, $S4$, $S5$, $S7$, $S19$, $S23$, $S20$, $S24$, $S25$, $S26$, $S37$,$S38$), NLP ($S20$, $S13$, $S12$, $S29$), QFD ($S17$, $S19$), CFV ($S27$), Rough Set Theory ($S21$, $S13$), IGA ($S19$, $S28$, $S32$), and CBRank ($S19$), RAR-P ($S33$), Middle Mile Optimization Platform ($S45$), and the

Table 4. Requirement representation in prioritization techniques

Requirements Representation	Prioritization Technique
Textual	360 Degree Feedback, Analytical Hierarchy Process, Enhanced Analytical Hierarchy Process, MoSCoW, Natural Language Processing
Formal	Black Hole Algorithm, Branch and Bound Algorithm, CBRank, Clustering (K-Means), Criticality Factor Value, FAGOSRA MS, Fuzzy Algorithm, Fuzzy C-mean, Improved Marine Predators Algorithm, Interactive Genetic Algorithm, Intuitionistic Fuzzy Sets, Knapsack, Logarithmic Fuzzy Trapezoidal Approach, Middle Mile Optimization Platform, Minimal Spanning Tree, Regression-based prioritization technique, Rough Set Theory, Rough-Fuzzy DEMATEL, Vertical Binary Search, Weighted Page Rank Method
Model-based	Goal Oriented Requirements Language, Social Network Analysis
Hybrid (Textual + Formal)	Active Learning and Ontological Modeling, Monitoring data and user feedback, Requirement Prediction Test-case Selection and Prioritization, Rule-Based Automated Requirements Prioritization, Software Quality Function Deployment

Table 5. Categories related to the limitations of requirements prioritization techniques

Category	Description
Applicability	Techniques have limitations when applied to different projects with varied contexts requiring flexible approaches.
Communication	Applying techniques in distributed teams can cause communication noise due to challenges such as different time zones, language barriers, and cultural differences.
Stakeholders Conflict	Techniques for requirements classification depend on the different perspectives of project members, which can lead to conflicts and disagreements.
Performance	Techniques that require high computational power to be executed efficiently or, when performed manually, result in high resource and time costs.
Scalability	Techniques suitable for handling large volumes of data are becoming inefficient in large-scale projects.
Human Factors	Techniques that heavily rely on human actions to be performed, which can introduce subjectivity into the results.
Requirement Interdependence	Techniques that need help to handle dependencies between requirements, where implementing one requirement depends on another, complicate the prioritization process.
Quality of Results	Techniques that may produce unsatisfactory or unclear results in the final processing require refinement of their outcomes.

Active Learning, and Ontology technique ($S44$). The dependency on human factors can vary according to perceptions, experiences, and business knowledge, potentially influencing the requirements prioritization process.

We grouped the K-Means technique ($S6$), 360 Degree Feedback ($S3$), Rough-Fuzzy DEMATEL ($S11$), IGA ($S19$, $S28$, $S32$), NLP ($S20$, $S13$, $S12$, $S29$), and Minimal Spanning Tree ($S25$) into **Interdependence between Requirements** category. Also included in this category are recent techniques such as RPTSP ($S39$), which struggles with ambiguity, redundancy, and frequent changes in requirements in agile environments; Social Network Analysis ($S40$), whose effectiveness depends on correctly identifying relationships between stakeholders and requirements, and the Black Hole Algorithm ($S43$), which only considers three types of dependencies (precedence, AND, XOR), ignoring more dynamic or complex relationships. Failing to address dependencies between requirements can lead to issues in planning and project execution. In the application of NLP techniques ($S20$, $S13$, $S12$, $S29$), if dependencies between requirements are not identified, the model may incorrectly prioritize requirements that depend on others that have not yet been implemented, resulting in a low-quality implementation that fails to satisfy the expectations.

Techniques such as the Weighted Page Rank Method ($S14$), Goal Oriented Requirements Language ($S16$), FAGOSRA MS ($S10$) and QFD ($S17$, $S19$) were associated with the category **Conflicts among Stakeholders**. The decisions made by various project members, including sponsors, developers, and product owners, significantly influence these techniques. Each member may have different views, priorities, and values assigned to each requirement, which generates conflicts during the prioritization process. The divergence in perspectives and interests of each project member can result in disagreements and difficulties in reaching a consensus, complicating the task of establishing a coherent prioritization of requirements by the real needs of the client/end users.

We associated the techniques Knapsack ($S31$), Regression-based Prioritization Technique ($S15$), and Minimal Spanning Tree ($S25$) with the category **Applicability** due to their limitations in terms of versatility for different types of software development projects. Techniques such as RAR-P ($S33$), RPTSP ($S39$), and Fuzzy-AHP ($S41$) also face applicability issues, either due to the need for specialized technical knowledge, limited validation contexts, or reduced sampling, which constrain their generalizability. Techniques based on monitoring and user feedback ($S36$) and the AHP method applied in a restricted geographical setting ($S42$) also highlight challenges in adapting to broader or dynamic project environments.

On the other hand, techniques such as AHP ($S1$, $S2$, $S4$, $S5$, $S7$, $S19$, $S23$, $S20$, $S24$, $S25$, $S26$), E-AHP ($S23$), Fuzzy Algorithm ($S18$, $S2$, $S21$), Improved Marine Predators Algorithm ($S34$), Vertical Binary Search ($S35$), and Active Learning, and Ontology ($S44$) face significant challenges in the categories of **Performance** and **Scalability**. These techniques are not recommended for projects involving large requirements due to their high computational cost or excessive resource consumption in the prioritization process, making them unsuitable for scenarios where scalability is a crucial concern.

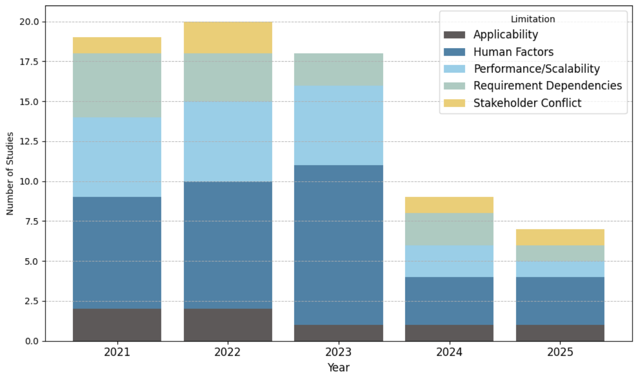

Figure 4. Distribution of Requirements Prioritization Techniques Limitations by Year

Analyzing data from 2021 to 2025 (Figure 4), the main challenges in prioritization techniques involve human factors and scalability, revealing reliance on subjective judgment and difficulty handling large requirement volumes. The limitation concerning requirement dependencies is also frequent, revealing gaps in the ability of techniques to manage complex relationships between functionalities. Applicability and stakeholder conflict appear more sporadically but still represent important barriers to practical adoption. These findings reinforce the need for more robust, automated, and context-adaptive methods for industrial environments.

Thus, we observe that the limitations of each technique underscore the importance of carefully considering the specific characteristics of each context and project when selecting and applying requirements prioritization techniques. We emphasize that the impact of the limitations identified in the analyzed techniques depends strongly on the context in which practitioners apply them. Scalability and performance issues, for example, are particularly critical in large-scale or data-intensive systems, such as enterprise resource planning (ERP) platforms or big data applications, where the volume of requirements tends to grow exponentially. Traditional techniques like the Analytic Hierarchy Process (AHP), despite their widespread adoption, exhibit scalability constraints. Study $S23$ discusses this limitation and introduces E-AHP as a more scalable alternative to the original method.

Conversely, limitations associated with stakeholder conflict or dependence on domain-specific knowledge are more prevalent in highly regulated environments, such as healthcare, aerospace, and public administration, where requirements are inherently complex, sensitive, and demand consensus among multiple stakeholders. Studies $S3$ and $S14$ propose approaches that effectively address prioritization challenges in such contexts by incorporating mechanisms for conflict resolution and stakeholder negotiation.

Moreover, techniques grounded in Artificial Intelligence and Machine Learning have aimed to mitigate human subjectivity in the prioritization process. However, these methods introduce new challenges, particularly concerning the reliability of input data and the complexity of integrating such techniques into agile development workflows. Studies $S29$ and $S33$ investigate the use of natural language processing (NLP) to extract and prioritize requirements automatically. At the same time, $S44$ presents a technique that leverages ontologies to bridge the semantic gap between user language and technical terminology.

In light of these observations, we recognize that the effectiveness of a prioritization technique depends not only on its methodological rigor but also on the characteristics of the target system and the constraints of its application domain. Practitioners must carefully consider these contextual factors when selecting and applying requirements prioritization approaches.

4.4 Listing of techniques for prioritizing software requirements evaluation scales (RQ3)

The listing describes how each technique is evaluated when applied. In this SLR, the evaluation scales used in each selected study were identified, as they influence how the techniques are assessed and compared. Table 6 presents the identified evaluation scales.

Among the scales identified in the studies are the nominal scale, which classifies data into unique categories without a defined order, facilitating the organization of data into groups, and the qualitative scale, which performs the evaluation based on characteristics or descriptions related to the quality of the results presented, providing a more subjective and descriptive analysis of the effectiveness of the techniques. During the study, we cataloged the scales according to their use in each study. In this SLR update, some new listing of scales were identified, in addition to those presented in [Rashdan, 2021], such as the statistical method of hypervolume and the qualitative method.

Twelve studies used the ratio scale to evaluate the techniques of the AHP ($S1$, $S4$, $S23$, $S24$,$S38$), QFD ($S19$), FAGOSRA MS ($S10$), Rough-Fuzzy DEMATEL ($S11$), Fuzzy Algorithm ($S2$) , Fuzzy-AHP ($S41$), Social Network Analysis($S40$) and the Black Hole Algorithm ($S43$) . This scale allows for an objective assessment of requirements through numerical measurements.

Fourteen studies used ordinal scales to evaluate the Regression-based prioritization ($S15$), NLP ($S20$,$S29$), Rough Set Theory ($S21$), Criticality Factor Value ($S27$), Fuzzy C-mean ($S9$), and QFD techniques ($S17$), RAR-P ($S33$), IMPA ($S34$), Vertical Binary Search ($S35$), Monitoring and feedback ($S36$), AHP ($S42$,$S37$), and Active Learning, and Ontology ($S44$). The ordinal scales employed in these studies allowed for ranking requirements based on

Table 6. Listing of prioritization scales

Scales	Study ID	#Papers
Ordinal	$S9$, $S15$, $S17$, $S20$, $S21$, $S27$, $S29$, $S33$, $S34$, $S35$, $S36$, $S37$, $S42$, $S44$	14
Ratio	$S1$, $S2$, $S4$, $S10$, $S11$, $S19$, $S23$, $S24$, $S38$, $S40$, $S41$, $S43$	12
Nominal	$S16$, $S18$, $S25$, $S7$	4
Statistical Method	$S14$, $S39$	2
Quantitative	$S6$, $S22$	2
Qualitative	$S26$	1
Interval	$S45$	1
Hyper-Volume Metric	$S30$	1
Not Mentioned	$S3$, $S5$, $S8$, $S12$, $S13$, $S28$, $S31$, $S32$	8

their importance, establishing a clear prioritization of requirements.

We identified that nominal scales were used to evaluate three techniques: Goal Oriented Requirements Language ($S16$), Fuzzy Algorithm ($S18$) and AHP ($S25$, $S7$). Nominal scales are essential for categorizing requirements without imposing a hierarchical order, allowing the classification of data into distinct groups based on specific characteristics. Techniques such as MoSCoW ($S6$) and K-Means ($S22$) were evaluated using quantitative scales, which offer a numerical and objective approach to requirements prioritization. Furthermore, we identified one study ($S45$) that employed an interval scale to evaluate the Middle Mile Optimization Platform technique.

We observed that techniques such as AHP and the Fuzzy Algorithm have been evaluated using multiple types of scales, including ratio, ordinal, and nominal, highlighting their flexibility and broad applicability across different software development scenarios. In addition, the Branch and Bound Algorithm ($S30$) technique was evaluated using the hypervolume metric to measure the quality of the results obtained by evaluating the application of multiple criteria in the efficient prioritization of requirements.

4.5 Steps involved in the process during the prioritization of software requirements (RQ4)

This RQ purpose is to identify a set of steps that play a specific role in the requirements prioritization process. We grouped the steps identified in the studies into four categories.

- C_1 - **Definition and Identification of Requirements:** The steps represent the initial phase of the prioritization process, involving gathering, classifying requirements, and identifying stakeholders.
- C_2 - **Structuring and Organization of Requirements:** The steps include structuring and organizing the elicited requirements, such as creating a backlog and grouping similar requirements.
- C_3 - **Classification and Evaluation of Requirements:** Steps necessary for the classification and prioritization of requirements, based on the criteria of each technique.
- C_4 - **Analysis and Decision about Requirements:** Steps related to validating and negotiating prioritized requirements according to the applied techniques.

In table 7 shows the steps involved in the Requirements Prioritization Process by studies. We grouped 93% (42/45) of the studies in the four categories.

Table 7. Distribution of studies by the steps categories involved in the requirements prioritization process

Steps	Study ID	#Papers
C_1	$S4$, $S5$, $S10$, $S13$, $S17$, $S20$, $S21$, $S24$, $S27$ $S34$, $S36$, $S37$, $S39$, $S40$, $S42$, $S44$, $S45$	17
C_2	$S2$, $S4$, $S5$, $S6$, $S8$, $S12$, $S13$, $S14$, $S15$, $S17$, $S23$, $S26$, $S30$, $S33$, $S34$, $S35$, $S36$, $S37$, $S38$, $S39$, $S40$, $S42$, $S43$, $S44$, $S45$	25
C_3	$S1$, $S2$, $S3$, $S7$, $S9$, $S11$, $S14$, $S15$, $S16$, $S18$, $S19$, $S22$, $S23$, $S24$, $S31$,$S33$, $S34$, $S35$, $S36$, $S37$, $S38$, $S39$, $S40$, $S41$, $S42$, $S43$, $S44$, $S45$	28
C_4	$S1$, $S4$, $S9$, $S10$, $S14$, $S23$, $S26$, $S28$,$S33$, $S34$, $S35$, $S36$, $S37$, $S38$, $S39$, $S40$, $S41$, $S42$, $S43$, $S44$, $S45$	21

In **C_1 - Definition and Identification of Requirements**, the step of defining the initial project requirements was mentioned in four articles ($S24$, $S27$, $S10$, $S20$). In contrast, the step of identifying stakeholders, or project sponsors, was described in five articles ($S17$, $S4$, $S5$, $S21$, $S13$). More recent studies also emphasized this phase: $S34$ discussed data acquisition from existing datasets; $S36$, $S39$, and $S40$ included requirement and stakeholder identification activities as part of their analysis and modeling processes; while $S37$, $S42$, $S44$, and $S45$ explicitly addressed the initial gathering or definition of requirements. Identifying and defining requirements and stakeholders is crucial to ensuring that all needs and expectations are met from the project's outset, providing a solid foundation for subsequent steps.

C_2 - Structuring and Organizing includes the creation of the product backlog, cited in 11 studies ($S17$, $S2$, $S4$, $S5$, $S26$, $S8$, $S23$, $S12$, $S13$, $S14$, $S15$). Additionally, the step of grouping requirements is essential for techniques that use Machine Learning, such as the E-AHP ($S23$) and in the use of the MoSCoW technique ($S6$)— identifying dependencies and interactions between requirements aids in the project's integrity and coherence, as mentioned in the study ($S30$). Reinforced this stage by incorporating organization and processing components prior to prioritization. For instance, S33 applied vectorization and clustering using K-Means; S34 and S39 grouped similar requirements based on dependency or prediction logic; S37 and S38 structured requirements within agile or modular units (e.g., Sprints or processing units); and studies like S44 and S45 employed feature extraction or functional clustering to prepare requirements for analysis. These steps ensure the organization of requirements, facilitating the implementation of techniques throughout the project.

We noted that techniques classified under C_3 **- Classification and Evaluation** of requirements heavily rely on human intervention to classify requirements. The step of Weight and Value Classification of Requirements is performed, as identified in 13 studies ($S1$, $S16$, $S18$, $S19$, $S2$, $S24$, $S3$, $S7$, $S9$, $S22$, $S23$, $S11$, $S14$). Additionally, we identified specific steps for each technique, such as uncertainty level classification ($S18$), requirement prediction using the Markov algorithm, which is essential for requirement evaluation ($S31$), and the filtering of valuable requirements ($S15$). Some studies further expanded this category by introducing diverse evaluation strategies, ($S33$) converted MoSCoW labels into numerical values to enable ranking, $S34$ and $S35$ calculated prioritization values based on stakeholder weights and dependency matrices, $S36$ used annotated user feedback to assign priority scores via decision matrices, $S37$ and $S42$ applied AHP-based comparison methods to rank backlog items, and techniques like $S41$, $S43$, $S44$, and $S45$ implemented fuzzy logic, simulation, optimization algorithms, or multi-criteria scoring to evaluate requirement importance.

We grouped 21 studies into the category C_4 **- Analysis and Decision**. The step of prioritization meetings with stakeholders is cited in two studies ($S1$, $S28$), emphasizing the importance of collaboration and consensus among stakeholders for informed decision-making. The creation of decision matrices, addressed in three studies ($S4$, $S9$, $S23$), provides a systematic framework for evaluating and comparing different requirements. Another step mentioned is the identification and resolution of requirement conflicts ($S26$, $S9$, $S10$). Additionally, the analysis of requirements from the customer's perspective, highlighted in study $S14$, ensures that the decisions made align with the sponsors' needs and expectations. Studies $S37$, $S41$, and $S42$ applied final selection and scoring steps and $S43$, $S44$, and $S45$ adopted decision support approaches such as simulation, optimization, and multi-criteria scoring to finalize requirements prioritization.

Some techniques were classified into more than one group, demonstrating various stages during their implementation. The E-AHP ($S23$) technique was classified into C_2, C_3, and C_4 categories. In C_2, the steps include the creation of a backlog and the grouping of similar requirements, in group C_3, the technique includes assigned weight and value to the backlog requirements, and in group C_4, a decision matrix was created. The Regression-based prioritization technique ($S15$) was classified into C_2 and C_3 categories. In C_2, the backlog creation step is mentioned, and in C_3 the filtering was done to consider only useful requirements. In the study, $S17$, the QFD technique was classified into C_1, with stakeholder identification, and into C_2, which applied the backlog creation stages. Other techniques such Fuzzy-AHP ($S41$) and Active Learning and Ontological Modeling ($S44$) were also assigned to multiple categories (e.g., C_2, C_3, and C_4), reflecting their integrated processes involving organization, evaluation, and final decision-making. Similarly, $S45$ combined functional clustering, weighted scoring, and prioritization illustrating a complete flow through C_1 to C_4.

4.6 Research type conducted in the studies (RQ5)

This RQ aims to identify the types of research used in the studies. We classified the studies according to the categories suggested by Wieringa *et al.* [2006], and for referencing purposes, we label them using the letter "T" (for Type) followed by a number:

- T_1 - **Evaluation Research:** focused on evaluating and measuring the effectiveness of a technique;
- T_2 - **Solution Proposal:** includes studies that propose new solutions;
- T_3 - **Philosophical Work:** applied in studies with theoretical discussions;
- T_4 - **Opinion Work:** consists of articles where the authors share their perspectives and interpretations;
- T_5 - **Experience Work:** refers to studies that report practical experiences and lessons learned.

The classification of the 45 studies into the categories T_1 (Evaluation Research) and T_2 (Solution Proposal) highlights significant trends in the research focus within the field (Table 8). The majority of the studies, 76%, fall into the T_2 category, indicating a strong emphasis on proposing new solutions. In contrast, 24% of the studies are classified under T_1 (Evaluation Research), which emphasizes the importance of assessing and measuring the effectiveness of techniques.

Table 8. Distribution of studies by research type

Research type	Study ID	#Papers
Solution Proposal	$S1$, $S3$, $S4$, $S5$, $S7$, $S8$, $S9$, $S10$, $S11$, $S12$, $S13$, $S14$, $S15$, $S16$, $S18$, $S19$, $S20$, $S21$, $S24$, $S25$, $S27$, $S28$, $S29$, $S31$, $S32$, $S33$, $S34$, $S35$, $S38$, $S40$, $S41$, $S42$, $S44$, $S45$	34
Evaluation Research	$S2$, $S6$, $S17$, $S22$, $S23$, $S26$, $S30$, $S36$, $S37$, $S39$, $S43$	11

The study $S20$ [Naufal Maulana and Siahaan, 2022] proposes a solution for requirements prioritization by combining the AHP technique with NLP based on conducted experiments. The study $S9$ by Ijaz *et al.* [2021] presents a new Fuzzy C-means technique based on Fuzzy algorithms, applying this approach to prioritizing non-functional requirements. The study $S28$ by Binti Rusli *et al.* [2023] introduces a solution proposal using the IGA technique, focusing on the dependencies between requirements. The study $S5$ by Somohano-Murrieta *et al.* [2021] conducts experiments to address the limitations of using the AHP technique, seeking to improve its application.

The study $S26$ by Wohlrab and Garlan [2023] evaluates the use of the AHP technique, focusing on the perspective of stakeholders. This study investigates how including stakeholders in the requirements prioritization process can influence the acceptance and effectiveness of the technique. The study

$S6$ by del Sagrado and del Águila [2021] assesses the application of the MoSCoW technique using clustering algorithms. This study uses advanced clustering methods to explore the combination of the MoSCoW technique, which categorizes requirements into four priorities (Must have, Should have, Could have, and Won't have). Finally, the study $S30$ by del Sagrado *et al.* [2023] evaluates the Branch and Bound technique in solving complex problems, such as defining the requirements to be implemented in the next delivery. Additionally, study $S43$ exemplifies recent efforts to evaluate the effectiveness, performance, and applicability of prioritization techniques in diverse and dynamic project contexts.

We can observe a predominance of studies classified as T_2, indicating that new research is focus on the development of new techniques for requirements prioritization and improvements to traditional techniques. The absence of studies in the other categories (T_3 Philosophical Work, T_4 Opinion Work, and T_5 Experience Work) points to a potential gap in the research landscape. Philosophical and opinion works can provide valuable insights and foster critical discussions that shape the direction of the field. Experience work, on the other hand, offers practical lessons and reflections that can guide future research and application.

4.7 Empirical strategies applied in the studies (RQ6)

This RQ focuses on the empirical studies conducted to evaluate the requirements prioritization techniques. For this, we used the Wohlin *et al.* [2012] guideline, which classify they as: (i) Experiment; (ii) Case study; (iii) Survey.

Table 9. Distribution of studies by empirical strategy

Empirical Strategy	Study ID	#Papers
Case Study	$S4$, $S7$, $S8$, $S9$, $S10$, $S11$, $S16$, $S18$, $S19$, $S21$, $S24$, $S25$, $S28$, $S37$, $S38$, $S39$, $S41$, $S43$, $S44$, $S45$	20
Experiment	$S1$, $S2$, $S5$, $S6$, $S12$, $S13$, $S14$, $S15$, $S17$, $S20$, $S22$, $S23$, $S27$, $S29$, $S30$, $S31$, $S32$, $S33$, $S36$, $S42$	20
Survey	$S26$	1
Not Mentioned	$S3$, $S34$, $S35$, $S40$	4

In Table 9, 41.6% of the studies were evaluated based on case studies. We noted that techniques were applied in real or simulated scenarios to analyze their performance in practical situations. In 41.6% of the studies, experiments were conducted. These experiments allow for a more rigorous and systematic analysis of the techniques in a controlled environment. Only 2.2% of the studies surveyed gathered experiences related to requirements prioritization techniques, and 14.5% did not provide sufficient information to validate the evaluation method used.

The relationship between research type and empirical is shown in Figure 5, highlighting the number of studies that identified them. The bubble chart in Figure 5 shows a summary of the empirical strategies addressed according to each category of the research types presented in Table 8.

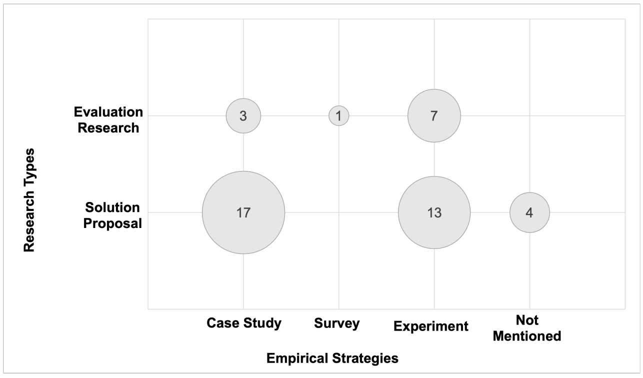

Figure 5. Distribution of empirical strategies according to the research types

The analysis of the 45 studies reveals an exciting overview of the intersection between research types and the empirical strategies employed. Among the studies analyzed, the predominance of the relationship between "solution proposal" and "case study" is notable, with 37.7% of studies falling into this category. On the other hand, 28.8% of studies related "solution proposal" to "experiment". This method allows for a more controlled and systematic analysis of new solutions, isolating variables and focusing on obtaining precise data on the techniques' performance.

Additionally, 6.6% of studies associated "research evaluation" with "experiment" highlighting the importance of testing and measuring the effectiveness of existing or modified techniques. Finally, we note that only 2.2% of studies related "research evaluation" to "survey", indicating that this combination is rare among the analyzed studies despite providing insights into the acceptance and effectiveness of requirements prioritization techniques from the perspective of users and stakeholders.

These results show a trend in applying and validating new solutions in real scenarios (case studies). At the same time, experiments are used to create new proposals and evaluate existing techniques. This dual approach ensures that new solutions are innovative and tested before widespread adoption. However, the low utilization of surveys suggests a potential area for future exploration. Incorporating more opinion-based research can enrich the understanding of how techniques are perceived and their real implications in the daily work of professionals involved in requirements prioritization. The importance of this empirical strategy cannot be overstated, as it can provide a more holistic and user-centered view in the development and evaluation of new techniques, instilling confidence in our research approach.

5 Discussion

This study aimed to update the state of the art on software requirements prioritization techniques, expanding the findings of the previous SLR conducted by Rashdan [Rashdan, 2021]. A total of 45 primary studies published between 2021 and June 2025 were selected and analyzed based on six research questions.

Regarding RQ1, we identified 32 different prioritization techniques, 23 of which are new compared to the previous review. Traditional techniques such as AHP, MoSCoW, and Fuzzy-based algorithms continue to be widely adopted (e.g., $S1$, $S2$, $S4$, $S6$, $S7$, $S18$). At the same time, more recent approaches based on Artificial Intelligence and Machine Learning, such as the Black Hole Algorithm ($S43$) and Active Learning with Ontological Modeling ($S44$), have gained traction. The E-AHP method, for instance, combines AHP with Machine Learning to overcome its classical limitations ($S23$).

These innovations reflect a trend toward greater automation and scalability in the prioritization process, however, the temporal analysis of the number of traditional and AI/ML-based techniques revealed a fluctuating adoption pattern between 2021 and June 2025, with a decline after 2022 and a slight recovery in 2025. Despite their innovative potential, AI/ML-based approaches still face significant barriers, while traditional techniques remain predominant-mainly due to their maturity and well-established integration in industrial environments.

In RQ2, the limitations of the techniques were grouped into eight categories, with particular emphasis on human factors (e.g., AHP in $S1$, $S2$, $S4$; MoSCoW in $S6$, $S22$; and NLP-based methods in $S12$, $S13$, $S20$, $S29$), requirement dependencies ($S6$, $S11$, $S19$, $S40$), and scalability issues ($S23$, $S35$, $S44$). Traditional techniques often struggle in large-scale or highly context-sensitive settings, especially when intense stakeholder interaction is required.

For RQ3, several assessment approaches were identified, including ordinal (e.g., $S9$, $S15$, $S21$), ratio (e.g., $S1$, $S2$, $S24$), and statistical-based methods (e.g., $S14$, $S39$). Although diverse, there is no clear standardization in how techniques are evaluated, which complicates comparisons across studies. Some techniques, such as Fuzzy-AHP ($S41$) and Branch and Bound ($S30$), were assessed using multiple approaches, demonstrating their flexibility but also highlighting the lack of consistent evaluation guidelines in the area.

Concerning RQ4, we identified four recurring stages in the prioritization process: requirements definition and identification (C1), organization (C2), classification (C3), and decision-making (C4). Most techniques encompass more than one of these steps. For instance, E-AHP (S23) and Active Learning ($S44$) support end-to-end prioritization processes, from requirements structuring to final decision. This suggests that modern approaches tend to be more integrated into broader requirement engineering workflows.

As for RQ5, most studies were classified as solution proposals (T2), accounting for 76% of the sample (e.g., $S5$, $S19$, $S33$). This predominance highlights the vitality of the area and the ongoing efforts toward innovation. However, the lack of philosophical, opinion-based, and experience report studies (T3 to T5) suggests a need for greater methodological diversity in the literature, including more reflective and practice-oriented perspectives.

Finally, for RQ6, most studies were evaluated through case studies (48.9%) or experiments (40%). Only one study employed a survey methodology ($S26$), and four studies did not report their empirical strategies. These findings indicate that practical validation is still limited, and more empirical investigations in real-world environments are needed.

Based on these findings, we identify the following research opportunities:

- **Context-sensitive techniques:** New methods should be designed to adapt to the specific characteristics of different domains and organizational contexts.
- **Standardization of evaluation criteria:** A unified framework for evaluating techniques could enhance consistency and comparability across studies.
- **Greater use of empirical research:** Techniques should be tested in real settings involving dynamic requirements and active stakeholder participation.
- **Exploration of GenAI-based approaches:** Generative AI may support tasks such as requirements analysis, clustering, and initial prioritization based on historical data.
- **Adoption of alternative research methods:** Philosophical reflections, opinion-based papers, and experience reports can broaden our understanding and foster stronger ties between academia and industry.

These findings complement and expand upon the results of prior secondary studies presented in Section 2, Table 1. For example, while R3 and R4 identified a limited number of techniques focused mainly on small and medium-sized requirement sets, our study uncovered a wider range of techniques, including 23 novel approaches, several of which are applicable to large-scale prioritization scenarios (e.g., $S30$, $S43$, $S44$). Additionally, unlike studies such as R2 and R6, which focused narrowly on AI-based techniques, our review offers a more comprehensive mapping that includes both traditional and AI-enhanced methods, allowing for a more balanced comparison. Regarding research types and validation strategies, our analysis also complements the findings of R7 by showing that despite increasing interest in Machine Learning applications, most proposals still lack robust empirical evaluation in real world settings. Overall, our study try to provide an updated and broader perspective on the evolution of the field and its current challenges.

In summary, this updated review reaffirms the relevance and complexity of software requirements prioritization, while also highlighting the progress and innovation that have emerged in recent years. By organizing the body of knowledge around key research questions, this study contributes to a clearer understanding of the strengths, limitations, and research gaps in the field. These insights may support both researchers in guiding future investigations and practitioners in selecting or designing prioritization techniques that better fit their project contexts.

6 Threats to Validity

SLR conducted in this study presents some threats to validity, which are categorized into internal validity and external validity.

Internal Validity. The application of the inclusion and exclusion criteria was conducted by the main author, and some subjectivity could have been embedded. In order to reduce this subjectivity, we held periodic meetings to clarify issues related to the selection and data extraction. All authors actively participated in these meetings, reviewing decisions and ensuring alignment throughout the process. Although the first author led the execution, the second and third authors were consistently involved in discussions, validations, and guidance for each stage of the review. This collaborative strategy helped mitigate individual bias and improve the reliability of the data analysis. Additionally, although strict criteria were applied, such as requiring that each study explicitly address at least one of the defined research questions, no formal quality assessment was conducted for the selected studies. This includes the absence of scoring schemes or checklist-based evaluations. This decision aimed to prioritize relevance and coverage. However, we recognize that future updates of this SLR could benefit from incorporating a more systematic evaluation of methodological quality.

External Validity. The conduct of this SLR update covered six different electronic research databases. In five of them, we had full access to the studies, while in the ACM database, our access was restricted only to public studies, without full access to private studies. This fact may have led to missing some primary studies. In addition, the decision to include only studies published in English may have resulted in the exclusion of significant studies published in other languages. Another limitation of this study is the absence of snowballing procedures, such as backward and forward reference checking. Although this technique could have increased the coverage of potentially relevant publications, we believe that the combination of a carefully designed search string and the use of six reputable digital libraries helped us capture a broad and representative sample. Moreover, the final set of 45 primary studies provides a comprehensive overview of the current landscape of requirements prioritization research.

7 Conclusions

In this paper, we have reported the results of an SLR update on requirements prioritization techniques. A total of 520 studies obtained from six databases were retrieved, of which only 45 persisted until the last selection stage. In general, we can summarize the findings of this research in: (i) 32 distinct techniques for requirements prioritization were found. (ii) Among the 32 techniques, 23 are new techniques presented for the first time in this review; (iii) AHP, Fuzzy Algorithms and NLP techniques stood out with the highest number of citations in the studies; (iv) Techniques based on ML and AI have demonstrated effectiveness in complex projects with large volumes of data; (v) The human factor was identified as the main challenge in applying prioritization techniques, impacting the objectivity of final decisions; (vi) The Ordinal Scale was the most commonly used evaluation methodology in studies on the application of techniques, closely followed by the Ratio Scale; (vii) Although each technique has its own methodology with specific implementation steps, some steps, such as creating a backlog and identifying stakeholders, overlap among the different techniques; (viii) Among the types of studies, most were identified as solution proposals, demonstrating the application of techniques in case studies.

The major contribution of the present work is to gather and explore the main requirements prioritization techniques from Rashdan's SLR update [Rashdan, 2021]. In addition, we believe that these SLR results can help identify a body of knowledge to support future research on new requirements prioritization techniques, providing a basis for other researchers and students who wish to learn about and contribute to this area. As future work, we intend to investigate the practical use of requirements prioritization techniques, comparing these results with their actual application in the software industry.

Declarations

Acknowledgements

Érica Ferreira de Souza is supported by CNPq (grant No. 404581/2025-0).

Authors' Contributions

Renato Cesar Ais contributed to the conceptualization, methodology, data collection, formal analysis, and writing the original draft preparation of this study. Érica Ferreira De Souza and Alinne C. Correa Souza contributed to the supervision, validation, and writing review and editing of the manuscript. All authors participated in the discussion of results, provided critical feedback to improve the paper, and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The dataset generated and analyzed during the current study is fully available in the Zenodo repository. The spreadsheet, which also served as a catalog to support data synthesis and ensure transparency and replicability, can be accessed at https://zenodo.org/records/16812333.

References

Achimugu, P., Selamat, A., and Ibrahim, R. (2014). A clustering based technique for large scale prioritization during requirements elicitation. In *Recent Advances on Soft Computing and Data Mining*, pages 623–632. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-07692-8_59.

Ahmad Al-Rawashdeh, T., Al'Azzeh, F., and Al-Tarawneh, F. (2025). An analytical hierarchy process-based technique for software requirements prioritization. *IEEE Access*, 13:50603–50610. DOI: https://doi.org/10.1109/ACCESS.2025.3552490.

Ahmed, M., Khan, S. U. R., and Alam, K. A. (2023). An NLP-based quality attributes extraction and prioritization framework in agile-driven software development. *Automated Software Engineering*, 30(1):7. DOI: https://doi.org/10.1007/s10515-022-00371-9.

Alaidaros, H., Bakodah, A., and Bamsaoud, S. F. (2022). A review on requirements prioritization approaches of software project management. In *2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, pages 1–5. DOI: https://doi.org/10.1109/ITSS-IoE56359.2022.9990930.

Ali, K., Khan, J. A., Aizaz, F., and Ahmed, M. (2021). Software requirements prioritization in the context of global software development. In *2021 International Conference on Frontiers of Information Technology (FIT)*, pages 13–18. DOI: https://doi.org/10.1109/FIT53504.2021.00013.

Almoqren, N. and Alrashoud, M. (2025). A smart framework for optimizing user feedback prioritization in application development. In *2024 International Conference on IT Inno-*

vation and Knowledge Discovery (ITIKD), pages 1–8. DOI: https://doi.org/10.1109/ITIKD63574.2025.11004934.

Álvarez, J. M. and Roibás-Millán, E. (2021). Agile methodologies applied to integrated concurrent engineering for spacecraft design. *Research in Engineering Design*, 32(4):431–450. DOI: https://doi.org/10.1007/s00163-021-00371-y.

Amelia, T. and Mohamed, R. (2023). A review: requirements prioritization criteria within collaboration perspective. *Pertanika Journal of Science & Technology*, 31(1). DOI: https://doi.org/10.47836/pjst.31.1.11.

Amelia, T. and Mohamed, R. B. (2021). A proposed requirements prioritization model based on cost-value approach with collaboration perspective. In *2021 International Conference on Software Engineering Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, pages 10–15. DOI: https://doi.org/10.1109/ICSECS52883.2021.00009.

Anwar, R. and Bashir, M. B. (2023). A systematic literature review of ai-based software requirements prioritization techniques. *IEEE Access*, 11:143815–143860. DOI: https://doi.org/10.1109/ACCESS.2023.3343252.

Armah, I. A., Hayfron-Acquah, J. B., Takyi, K., Gyening, R.-M. O. M., and Eshun, M. (2023). The use of knapsack 0/1 in prioritizing software requirements and markov chain to predict software success. *International Journal of Information Technology*, 16(3):1967–1979. DOI: https://doi.org/10.1007/s41870-023-01476-4.

Bai, S., Shi, S., Han, C., Yang, M., Gupta, B. B., and Arya, V. (2024). Prioritizing user requirements for digital products using explainable artificial intelligence: A data-driven analysis on video conferencing apps. *Future Generation Computer Systems*, 158:167–182. DOI: https://doi.org/10.1016/j.future.2024.04.037.

Binkhonain, M. and Zhao, L. (2023). A machine learning approach for hierarchical classification of software requirements. *Machine Learning with Applications*, 12:100457. DOI: https://doi.org/10.1016/j.mlwa.2023.100457.

Binti Rusli, S. N., Abu Bakar, R. B., and Binti Abdul Hamid, S. S. (2023). An improvement of interactive prioritization technique for requirements interdependency in prioritization process. In *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, pages 320–325. DOI: https://doi.org/10.1109/ICSECS58457.2023.10256396.

Bourque, P., Fairley, R. E., and Society, I. C. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0.* IEEE Computer Society Press, Washington, DC, USA, 3rd edition.

Brahmam, M. G., Anand R, V., Grover, V., and Gochhait, S. (2024). Optimizing requirements prioritization: Majority voting goal-based approach with vertical binary search. In *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, pages 1283–1288. DOI: https://doi.org/10.1109/ICETSIS61505.2024.10459533.

Cao, L. and Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software*, 25(1):60–67. DOI: https://doi.org/10.1109/MS.2008.1.

Chen, C., He, Y., and Bai, S. (2024). A software requirement prioritization method for online education software development. In *2024 4th International Conference on Information Communication and Software Engineering (ICICSE)*, pages 25–29. DOI: https://doi.org/10.1109/ICICSE61805.2024.10625686.

del Sagrado, J. and del Águila, I. M. (2021). Assisted requirements selection by clustering. *Requirements Engineering*, 26(2):167–184. DOI: https://doi.org/10.1007/s00766-020-00341-1.

del Sagrado, J., Sierra Ibañez, J. A., and del Águila, I. M. (2023). An estimation of distribution algorithm based on interactions between requirements to solve the bi-objective next release problem. *Journal of Systems and Software*, 199:111632. DOI: https://doi.org/10.1016/j.jss.2023.111632.

Fadel, S., Abdelmoez, W., and Saad, A. A. (2022). Considering multiple stakeholders perspectives for interval-based goal oriented requirements prioritization in agile development. In *2022 6th International Conference on Computer, Software and Modeling (ICCSM)*, pages 14–21. DOI: https://doi.org/10.1109/ICCSM57214.2022.00010.

Garner, P., Hopewell, S., Chandler, J., MacLehose, H., Akl, E. A., Beyene, J., Chang, S., Churchill, R., Dearness, K., Guyatt, G., Lefebvre, C., Liles, B., Marshall, R., Martínez García, L., Mavergames, C., Nasser, M., Qaseem, A., Sampson, M., Soares-Weiser, K., Takwoingi, Y., Thabane, L., Trivella, M., Tugwell, P., Welsh, E., Wilson, E. C., and Schünemann, H. J. (2016). When and how to update systematic reviews: consensus and checklist. *BMJ*, 354. DOI: https://doi.org/10.1136/bmj.i3507.

Gerogiannis, V., K. Kitsis, N., Tzimos, D., and Son, L. H. (2022). A novel requirements prioritization approach based on 360 degree feedback and group recommendation. In *Proceedings of the 5th International Conference on Algorithms, Computing and Systems*, ICACS '21, pages 48–54, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3490700.3490709.

Gupta, A. and Gupta, C. (2022). A novel collaborative requirement prioritization approach to handle priority vagueness and inter-relationships. *Journal of King Saud University - Computer and Information Sciences*, 34(5):2288–2297. DOI: https://doi.org/10.1016/j.jksuci.2019.12.002.

Hussain, A. and Mkpojiogu, E. O. C. (2016). Requirements: Towards an understanding on why software projects fail. *AIP Conference Proceedings*, 1761(1):020046. DOI: https://doi.org/10.1063/1.4960886.

Ibrahim Alfassam, N., Abdullah-Al-Wadud, M., and Rashed Alrashoud, M. (2025). Black hole algorithm for software requirements prioritization. *IEEE Access*, 13:95810–95820. DOI: https://doi.org/10.1109/ACCESS.2025.3574998.

Ijaz, K. B., Inayat, I., Daneva, M., and Bukhsh, F. A. (2021). Value based fuzzy approach for non-functional requirements prioritization. In Shishkov, B., editor, *Business Modeling and Software Design*, pages 330–342, Cham. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-79976-2_21.

Ilays, I., Hafeez, Y., Almashfi, N., Ali, S., Humayun, M., Aqib, M., and Alwakid, G. (2024). Towards improving the quality of requirement and testing process in agile software development: An empirical study. *Computers, Materials & Continua*, 80(3):3761–3784. DOI: https://doi.org/10.32604/cmc.2024.053830.

Izhar, R., Cosh, K., and Bhatti, S. N. (2024). Enhancing agile software development: A novel approach to automated requirements prioritization. In *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 286–293. DOI: https://doi.org/10.1109/JCSSE61278.2024.10613648.

Kaleem, S., Asim, M., El-Affendi, M., and Babar, M. (2024). Optimizing requirements prioritization for iot applications using extended analytical hierarchical process and an advanced grouping framework. *Future Internet*, 16(5):160. DOI: https://doi.org/10.3390/fi16050160.

Karlsson, J. and Ryan, K. (1997). A cost-value approach for prioritizing requirements. *IEEE Software*, 14(5):67–74. DOI: https://doi.org/10.1109/52.605933.

Karlsson, J., Wohlin, C., and Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14):939–947. DOI: https://doi.org/10.1016/S0950-5849(97)00053-0.

Keele, S. *et al*. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, ver. 2.3 ebse technical report. ebse.

Kifetew, F. M., Perini, A., Susi, A., Siena, A., Munante, D., and Morales-Ramirez, I. (2021). Automating user-feedback driven requirements prioritization. *Information and Software Technology*, 138:106635. DOI: https://doi.org/10.1016/j.infsof.2021.106635.

Kitchenham, B. A. (2012). Systematic review in software engineering: Where we are and where we should be going. In *Proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies*, pages 1–2. Association for Computing Machinery. DOI: https://doi.org/10.1145/2372233.2372235.

Klievtsova, N., Mangler, J., Kampik, T., and Rinderle-Ma, S. (2024). Utilizing process models in the requirements engineering process through model2text transformation. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 205–217. DOI: https://doi.org/10.1109/RE59067.2024.00028.

Krishnan, S. K., T, N., and R, K. (2023). Systematic literature review of product feature prioritization frameworks in startups building digital products using prisma. In *2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 1137–1143. DOI: https://doi.org/10.1109/R10-HTC57504.2023.10461802.

Malgaonkar, S., Licorish, S. A., and Savarimuthu, B. T. R. (2022). Prioritizing user concerns in app reviews – a study of requests for new features, enhancements and bug fixes. *Information and Software Technology*, 144:106798. DOI: https://doi.org/10.1016/j.infsof.2021.106798.

Martinis, A., Tzimos, D., Gerogiannis, V. C., and Hoang Son, L. (2022). A multiple stakeholders' software requirements prioritization approach based on intuitionistic fuzzy sets. In *2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, pages 1–5. DOI: https://doi.org/10.1109/CTISC54888.2022.9849773.

Masadeh, R., Alzaqebah, A., and Hudaib, A. (2018). Grey wolf algorithm for requirements prioritization. *Modern Applied Science*, 12. DOI: https://doi.org/10.5539/mas.v12n2p54.

Mendes, E., Wohlin, C., Felizardo, K., and Kalinowski, M. (2020). When to update systematic literature reviews in software engineering. *Journal of Systems and Software*, 167:110607. DOI: https://doi.org/10.1016/j.jss.2020.110607.

Miranda, E. (2022). Moscow rules: A quantitative exposé. In *Agile Processes in Software Engineering and Extreme Programming*, pages 19–34, Cham. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-031-08169-9_2.

Model, K., Mombrey, C., and Herzwurm, G. (2022). Paving the way to a software supported requirements prioritization in distributed scrum projects. In *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)*, pages 51–58. DOI: https://doi.org/10.1145/3524614.3528633.

Mohamed, N., Mazen, S., and Helmy, W. (2022). E-ahp: An enhanced analytical hierarchy process algorithm for prioritizing large software requirements numbers. *International Journal of Advanced Computer Science and Applications*, 13(7):190–199. DOI: https://doi.org/10.14569/IJACSA.2022.0130725.

Mohammad, C. W., Shahid, M., and Hussain, S. Z. (2021). Fuzzy attributed goal oriented software requirements analysis with multiple stakeholders. *International Journal of Information Technology*, 13(6):1–9. DOI: https://doi.org/10.1007/s41870-017-0073-0.

Monteiro, M. d. S. and Batista, G. O. d. S. (2023). Investigating usability pitfalls in brazilian and foreign governmental chatbots. *Journal on Interactive Systems*, 14(1):331–340. DOI: https://doi.org/10.5753/jis.2023.3104.

Muhammad, A., Siddique, A., Mubasher, M., Aldweesh, A., and Naveed, Q. N. (2023). Prioritizing non-functional requirements in agile process using multi criteria decision making analysis. *IEEE Access*, 11:24631–24654. DOI: https://doi.org/10.1109/ACCESS.2023.3253771.

Naufal Maulana, M. Z. and Siahaan, D. (2022). Use case-based analytical hierarchy process method for software requirements prioritization. In *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 205–210. DOI: https://doi.org/10.1109/ICITISEE57756.2022.10057944.

Nazim, M., Mohammad, C. W., and Sadiq, M. (2022). Fuzzy-based methods for the selection and prioritization of software requirements: a systematic literature review. In *Evolution in Computational Intelligence: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)*, pages 115–129. Springer. DOI: https://doi.org/10.1007/978-981-16-6616-2_11.

Perini, A., Susi, A., Ricca, F., and Bazzanella, C. (2007). An empirical study to compare the accuracy of ahp and cbranking techniques for requirements prioritization. In *Fifth International Workshop on Comparative Evalua-*

tion in Requirements Engineering, pages 23–35. DOI: https://doi.org/10.1109/CERE.2007.1.

Pressman, R. S. and Maxim, B. R. (2014). *Software engineering: A practitioner's approach*. McGraw-Hill Higher Education, 8th edition.

Rashdan, A. (2021). Requirement prioritization in software engineering: A systematic literature review on techniques and methods. Dissertation, Linnaeus University, Faculty of Technology, Department of computer science and media technology (CM).

Rehman Khan, S. U., Younus, M., Iqbal, J., and Basit Ur Rahim, M. A. (2024). A fuzzy ahp-based quantitative framework to prioritize the crowd-based requirements. In *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, pages 680–691. DOI: https://doi.org/10.1109/QRS-C63300.2024.00090.

Reinehr, S. (2020). *Engenharia de requisitos*. Porto Alegre: SAGAH, 1 edition.

Ribeiro, J. P. and Garcés, L. (2023). Especificação de requisitos de design de software para sistemas de iot conforme a lgpd: Resultados de aplicação em um sistema de assistência para pacientes com diabetes mellitus. In *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*, pages 37–42. SBC. DOI: https://doi.org/10.5753/sbcas_estendido.2023.229693.

Rojas, L., Olivares-Rodríguez, C., Alvarez, C., and Campos, P. G. (2022). Ourrank: A software requirements prioritization method based on qualitative assessment and cost-benefit prediction. *IEEE Access*, 10:131772–131787. DOI: https://doi.org/10.1109/ACCESS.2022.3230152.

Sadiq, M. and Devi, V. S. (2022). A rough-set based approach for the prioritization of software requirements. *International Journal of Information Technology*, 14(1):447–457. DOI: https://doi.org/10.1007/s41870-021-00749-0.

Sadiq, M. and Devi, V. S. (2023). Prioritization and selection of the software requirements using rough-set theory. *IETE Journal of Research*, 69(8):5169–5186. DOI: https://doi.org/10.1080/03772063.2021.1973593.

Sadiq, M., Sadim, M., and Parveen, A. (2021). Applying statistical approach to check the consistency of pairwise comparison matrices during software requirements prioritization process. *International Journal of System Assurance Engineering and Management*, 12(3):451–460. DOI: https://doi.org/10.1007/s13198-021-01090-2.

Singh, Y. V., Kumar, B., and Chand, S. (2021). Requirements prioritization using logarithmic fuzzy trapezoidal approach (lfta). In *Innovations in Information and Communication Technologies (IICT-2020)*, pages 309–318, Cham. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-66218-9_36.

Sommerville, I. (2011). *Software Engineering*. Pearson Addison-Wesley, 9 edition.

Somohano-Murrieta, J. C. B., Ocharán-Hernández, J. O., Ángel J. Sánchez-García, Limón, X., and de los Ángeles Arenas-Valdés, M. (2021). Improving the analytic hierarchy process for requirements prioritization using evolutionary computing. *Programming and Computer Software*, 47(8):746–756. DOI: https://doi.org/10.1134/S0361768821080235.

Talele, P. and Phalnikar, R. (2021). Classification and prioritisation of software requirements using machine learning- a systematic review. In *2021 11th international conference on cloud computing, data science & engineering (confluence)*, pages 912–918. IEEE. DOI: https://doi.org/10.1109/Confluence51648.2021.9377190.

Tanveer, A., Ibrahim, N. B., Rehman, M. Z., and Nawi, N. M. (2024). A framework for handling scalability in requirements prioritization using impa algorithm for large scale projects. In *2024 2nd International Conference on Computing and Data Analytics (ICCDA)*, pages 1–5. DOI: https://doi.org/10.1109/ICCDA64887.2024.10867359.

Tanveer, S. S. and Rana, Z. A. (2024). Prioritizing software requirements by combining the usage monitoring and user feedback data. *IEEE Access*, 12:82825–82841. DOI: https://doi.org/10.1109/ACCESS.2024.3409847.

Turkmen, A. D., Gokdag, Z. H., and Cebeci, S. (2025). Product strategy management using business process modelling for middle mile operations. *Procedia Computer Science*, 256:1813–1820. DOI: https://doi.org/10.1016/j.procs.2025.02.322.

Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria a proposal and a discussion. *Requirements Engineering*, 11(1):102–107. DOI: https://doi.org/10.1007/s00766-005-0021-6.

Winton, J. and Palma, F. (2023). Improving software requirements prioritization through the lens of constraint solving. *arXiv preprint arXiv:2306.12391*. DOI: https://doi.org/10.48550/arXiv.2306.12391.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., *et al.* (2012). *Experimentation in software engineering*, volume 236. Springer.

Wohlrab, R. and Garlan, D. (2023). A negotiation support system for defining utility functions for multi-stakeholder self-adaptive systems. *Requirements Engineering*, 28(1):3–22. DOI: https://doi.org/10.1007/s00766-021-00368-y.

Yaseen, M. (2023). Exploratory study of existing research on software requirements prioritization: A systematic literature review. *Journal of Software: Evolution and Process*, 36(6):e2613. DOI: https://doi.org/10.1002/smr.2613.

Yaseen, M., Mustapha, A., Shah, M. A., and Ibrahim, N. (2023). A hybrid technique using minimal spanning tree and analytic hierarchical process to prioritize functional requirements for parallel software development. *Requirements Engineering*, 28(3):347–376. DOI: https://doi.org/10.1007/s00766-023-00397-9.

Zhang, X., Li, J., Eres, H., and Zheng, C. (2021). Prioritizing and aggregating interacting requirements for product-service system development. *Expert Systems with Applications*, 185. DOI: https://doi.org/10.1016/j.eswa.2021.115636.

A Selected Studies

Table 10. List of the 45 studies included in the updated SLR

ID	Ref.	Title	Year	Venue Type	Evaluation
S1	Ali *et al*. [2021]	Software Requirements Prioritization in the context of Global Software Development	2021	Conference	Experiment
S2	Amelia and Mohamed [2021]	A Proposed Requirements Prioritization Model Based on Cost-Value Approach with Collaboration Perspective	2021	Conference	Experiment
S3	Gerogiannis *et al*. [2022]	A Novel Requirements Prioritization Approach based on 360 Degree Feedback and Group Recommendation	2022	Conference	Not mentioned
S4	Sadiq *et al*. [2021]	Applying statistical approach to check the consistency of pairwise comparison matrices during software requirements prioritization process	2021	Journal	Case Study
S5	Somohano-Murrieta *et al*. [2021]	Improving the Analytic Hierarchy Process for Requirements Prioritization Using Evolutionary Computing	2021	Journal	Experiment
S6	del Sagrado and del Águila [2021]	Assisted requirements selection by clustering	2021	Journal	Experiment
S7	Álvarez and Roibás-Millán [2021]	Agile methodologies applied to Integrated Concurrent Engineering for spacecraft design	2021	Journal	Case Study
S8	Singh *et al*. [2021]	Requirements Prioritization Using Logarithmic Fuzzy Trapezoidal Approach (LFTA)	2021	Conference	Case Study
S9	Ijaz *et al*. [2021]	Value-Based Fuzzy Approach for Non-functional Requirements Prioritization	2021	Conference	Case Study
S10	Mohammad *et al*. [2021]	Fuzzy attributed goal oriented software requirements analysis with multiple stakeholders	2021	Journal	Case Study
S11	Zhang *et al*. [2021]	Prioritizing and aggregating interacting requirements for product-service system development	2021	Journal	Case Study
S12	Kifetew *et al*. [2021]	Automating user-feedback driven requirements prioritization	2021	Journal	Experiment
S13	Sadiq and Devi [2023]	Prioritization and Selection of the Software Requirements using Rough-Set Theory	2023	Journal	Experiment
S14	Gupta and Gupta [2022]	A novel collaborative requirement prioritization approach to handle priority vagueness and inter-relationships	2021	Journal	Experiment
S15	Malgaonkar *et al*. [2022]	Prioritizing user concerns in app reviews – A study of requests for new features, enhancements and bug fixes	2021	Journal	Experiment
S16	Fadel *et al*. [2022]	Considering Multiple Stakeholders Perspectives for interval-based Goal Oriented Requirements Prioritization in agile development	2022	Conference	Case Study
S17	Model *et al*. [2022]	Paving the Way to a Software-Supported Requirements Prioritization in Distributed Scrum Projects	2022	Conference	Experiment
S18	Martinis *et al*. [2022]	A Multiple Stakeholders Software Requirements Prioritization Approach based on Intuitionistic Fuzzy Sets	2022	Conference	Case Study
S19	Rojas *et al*. [2022]	OurRank: A Software Requirements Prioritization Method Based on Qualitative Assessment and Cost-Benefit Prediction	2022	Journal	Case Study
S20	Naufal Maulana and Siahaan [2022]	Use Case-Based Analytical Hierarchy Process Method for Software Requirements Prioritization	2022	Conference	Experiment
S21	Sadiq and Devi [2022]	A rough-set based approach for the prioritization of software requirements	2022	Journal	Case Study
S22	Miranda [2022]	Moscow Rules: A Quantitative Exposé	2022	Conference	Experiment

(Continued on next page)

Table 10. List of sources selected in the SLR (continued)

ID	Ref.	Title	Year	Venue Type	Evaluation
S23	Mohamed *et al*. [2022]	E-AHP: An Enhanced Analytical Hierarchy Process Algorithm for Priotrizing Large Software Requirements Numbers	2022	Journal	Experiment
S24	Muhammad *et al*. [2023]	Prioritizing Non-Functional Requirements in Agile Process Using Multi Criteria Decision Making Analysis	2023	Journal	Case Study
S25	Yaseen *et al*. [2023]	A hybrid technique using minimal spanning tree and analytic hierarchical process to prioritize functional requirements for parallel software development	2023	Journal	Case Study
S26	Wohlrab and Garlan [2023]	A negotiation support system for defining utility functions for multi-stakeholder self-adaptive systems	2023	Journal	Survey
S27	Ahmed *et al*. [2023]	An NLP-based quality attributes extraction and prioritization framework in Agile-driven software development	2023	Journal	Experiment
S28	Binti Rusli *et al*. [2023]	An Improvement of Interactive Priorization Technique for Requirements Interdependency in Prioritization Process	2023	Conference	Case Study
S29	Binkhonain and Zhao [2023]	A machine learning approach for hierarchical classification of software requirements	2023	Journal	Experiment
S30	del Sagrado *et al*. [2023]	An estimation of distribution algorithm based on interactions between requirements to solve the bi-objective Next Release Problem	2023	Journal	Experiment
S31	Armah *et al*. [2023]	The use of knapsack 0/1 in prioritizing software requirements and Markov chain to predict software success	2023	Journal	Experiment
S32	Winton and Palma [2023]	Improving Software Requirements Prioritization through the Lens of Constraint Solving	2023	Conference	Experiment
S33	Izhar *et al*. [2024]	Enhancing Agile Software Development: A Novel Approach to Automated Requirements Prioritization	2024	Conference	Experiment
S34	Tanveer *et al*. [2024]	A Framework for Handling Scalability in Requirements Prioritization Using IMPA Algorithm for Large Scale Projects	2024	Conference	Not mentioned
S35	Brahmam *et al*. [2024]	Optimizing Requirements Prioritization: Majority Voting Goal-Based Approach with Vertical Binary Search	2024	Conference	Not mentioned
S36	Tanveer and Rana [2024]	Prioritizing Software Requirements by Combining the Usage Monitoring and User Feedback Data	2024	Journal	Experiment
S37	Chen *et al*. [2024]	A Software Requirement Prioritization Method for Online Education Software Development	2024	Conference	Case Study
S38	Kaleem *et al*. [2024]	Optimizing Requirements Prioritization for IoT Applications Using Extended Analytical Hierarchical Process and an Advanced Grouping Framework	2024	Journal	Case Study
S39	Ilays *et al*. [2024]	Towards Improving the Quality of Requirement and Testing Process in Agile Software Development: An Empirical Study	2024	Journal	Case Study
S40	Bai *et al*. [2024]	Prioritizing user requirements for digital products using explainable artificial intelligence: A data-driven analysis on video conferencing apps	2024	Journal	Not mentioned
S41	Rehman Khan *et al*. [2024]	A Fuzzy AHP-based Quantitative Framework to Prioritize the Crowd-Based Requirements	2024	Conference	Case Study

(Continued on next page)

Table 10. List of sources selected in the SLR (continued)

ID	Ref.	Title	Year	Venue Type	Evaluation
S42	Ahmad Al-Rawashdeh *et al*. [2025]	An Analytical Hierarchy Process-Based Technique for Software Requirements Prioritization	2025	Journal	Experiment
S43	Ibrahim Alfassam *et al*. [2025]	Black Hole Algorithm for Software Requirements Prioritization	2025	Journal	Case Study
S44	Almoqren and Alrashoud [2025]	A Smart Framework for Optimizing User Feedback Prioritization in Application Development	2025	Conference	Case Study
S45	Turkmen *et al*. [2025]	Product strategy management using business process modeling for middle mile operations	2025	Conference	Case Study