Online-Journals.org

iTDAF IETI Transactions on Data Analysis and Forecasting

iTDAF | eISSN: 2959-0442 | Vol. 3 No. 4 (2025) | OPEN ACCESS

https://doi.org/10.3991/itdaf.v3i4.59215

PAPER

Introducing a Framework for Requirement Prioritization Using Multi-Aspect Decision Making

Zara Khalid[1](✉), Inam Ullah[1], Sidra Tahir[1], Sadia Ali[1], Ahtasham Sajid[2], Sabitha Banu[3]

[1]PMAS – Arid Agriculture University, Rawalpindi, Pakistan

[2]Air University, Islamabad, Pakistan

[3]PSGR Krishnammal College for Women, Coimbatore, India

zarakhalid571@gmail.com

ABSTRACT

In software development, it is crucial to highlight the most important requirements related to software components. The process of requirement elicitation and prioritization actively involved stakeholders, requiring several decisions to meet their requirements. Often useful requirement components are neglected. This paper proposes a framework that is meant to promote multi-faceted decision-making based on a number of dimensions such as business values, stakeholder preference, functional constraints, conflicts between requirements, and requirements' interdependencies and thereby improve the quality and efficiency of the requirement prioritization process. The framework serves to enrich the functional attributes of requirements. Through a detailed literature review, this research found aspects of requirements that have the most influence on the software project and the business process. Moreover, the framework utilizes natural language processing (NLP) methods in the ability to identify conflicts between stakeholders' needs and to ensure requirements' completeness and clarity. The combination of multi-faceted analysis together with stakeholder priority analysis improves the requirement prioritization process. The results show that the proposed framework successfully reduces the possibility of useful requirement components being overlooked, lowers conflicts between stakeholders, and improves requirements' completeness. This approach improves the decision-making process associated with prioritization by allowing requirement analysis to be done from various dimensions. Moreover, it pinpoints possible success factors related to software requirements and thereby facilitates the software engineering team in effectively prioritizing and implementing these requirements and thereby improving the quality of the software engineering process.

KEYWORDS

requirement prioritization, requirement conflicts, requirement uncertainty, multi-aspect decision-making

Khalid, Z., Ullah, I., Tahir, S., Ali, S., Sajid, A., Banu, S. (2025). Introducing a Framework for Requirement Prioritization Using Multi-Aspect Decision Making. *IETI Transactions on Data Analysis and Forecasting (iTDAF)*, 3(4), pp. 55–69. https://doi.org/10.3991/itdaf.v3i4.59215

Article submitted 2025-07-13. Revision uploaded 2025-09-23. Final acceptance 2025-09-26.

1 INTRODUCTION

The penetration of software automation in recent years has affected many fields of life. The software development life cycle recognizes requirement engineering as an essential and dominant element. The realization of successful software development requires careful consideration of functional and non-functional requirements because both of them altogether impact the behavior of the system, performance, usability, dependability, and overall feasibility in terms of meeting stakeholder expectations and project objectives [1]. The ever-growing need for automation in almost all fields of modern life has made software an integral element with the effect of boosting productivity, supporting intelligent decisions, promoting ease of operation, and allowing the integration of everyday activities [2]. Practitioners, software engineers, and researchers often face several difficulties throughout the processes of requirement management and priority. A software product or project contains many requirements, which require appropriate organization and detailing. The entire process must be carried out successfully in order to please stakeholders' expectations and meet the quality requirements. Requirement prioritization involves decision-making based on different factors, i.e., time, cost, risk, workforce, and other factors [3].

Customer satisfaction is greatly influenced by the quality of the software project. Successful software product development needs consistent requirements and expert people with good knowledge of product line engineering. Inconsistent requirements may result in variations that do not satisfy the stakeholder needs within the software product. Effective implementation of a particular software under consideration relies on numerous critical aspects, such as understanding customer inclinations, optimizing asset utilization, overseeing delivery timelines, guaranteeing stakeholder collaboration, and keeping up with overall quality through the development lifecycle [4]. Based on significance, requiring meeting quality attributes is considered an important part of the requirement engineering phase [5]. Software requirement prioritization and selection serve as a building block to produce a quality software product or project [6]. Prioritization techniques will help in better elaboration and understanding of stakeholder requirements to get more information out of them. A natural language processing (NLP) model is used to extract information from customer requirements and rank them based on their overall impact [7].

Traditional prioritization methods were frequently incapable of providing requirement priorities that completely meet the stakeholder expectations, resulting in gaps between stakeholder needs, decision-making exactness, and, in general, project alignment and adequacy [8]. In the software development life cycle, it is most crucial to gather, assess, and prioritize the requirements in a structured way while keeping their impacts in mind for the specific project. Although it is crucial to satisfy the stakeholders' needs, the stakeholders also have divergent views towards each requirement. Requirement engineering practice has a major impact on the success or failure of a software project in bringing clarity, alignment, achievability, and sound choices for the duration of software development that benefits the end quality of the project and the final outcome [9]. With the scenario of the analysis of requirements, the requirement prioritization is assumed to be a preliminary stage in light of constraints for the project's resources [10]. Maintaining a huge set of requirements gets to be complex and difficult during the prioritization process [11].

Requirement prioritization helps identify and prioritize the most significant requirements based on defined prioritization criteria to guarantee optimal decision-making, productive resource allocation, and alignment with stakeholder expectations, eventually improving software project planning, quality, and overall success [12]. Traditional requirement prioritization techniques are primarily human-driven, depending on expert judgment and specialized analysis; in any case, many of these approaches stay one-dimensional. Culture-based prioritization techniques emphasize straightforwardness within the software product. Generally, traditional techniques fail to identify problems encountered by stakeholders [13]. Success requirements prioritization for a particular software system improves proficiency and increases the probability of delivering an effective product that satisfies stakeholder expectations. The proposed framework focuses on multiple objectives as mentioned below:

- To increase the effectiveness of decision-making during the requirement engineering phase.
- To increase requirements consistency meeting larger organization plans and goals.
- To minimize stakeholder conflicts for better software development.

The remaining part of the paper is organized as follows: Section 2 consists of related work, a review of existing prioritization techniques. Section 3 briefly explain the working of the proposed framework, Section 4 presents the results and discussions, and finally Section 5 contains future work.

2 LITERATURE REVIEW

Software product development success depends on increased customer satisfaction. Incompetent requirement prioritization management may lead to problems, i.e., decreased customer satisfaction, increased development cost and time, and software product inconsistencies. A study focused on the importance of stakeholder identification was conducted. The proposed approach used fuzzy set theory and social networks to handle the requirement uncertainty and ambiguity problem. Compared with existing approaches, it was applied to the examination system to measure applicability in terms of requirement elicitation and prioritization [14]. A research study discussed the problems with current prioritization methods, which may lead to the selection of inaccurate priorities and being less reliable. To better understand these problems, critical success factors were identified, and experts' opinions on these factors were obtained. The analytical hierarchy process (AHP) used for the prioritization purpose also pointed out that future research needs to focus on assessing the social factors' impact on deciding which requirements are most important [5].

Prioritization criteria greatly contributed to the overall process. But an approach highlighted that prioritization criteria may vary during different phases with the change of requirements. It was concluded that prioritization criteria are not equal all the time, and future directions for conducting more quantitative studies on varied projects and contexts to understand the impact of requirement prioritization criteria on decision-making were suggested [15]. Multiple stakeholders were involved in a software project, and they have diverse perspectives on each requirement. To address this, a multiple-perspective prioritization technique algorithm is proposed for enhancing software requirements prioritization activity. A controlled experiment was conducted involving 159 participants. However, the study utilized a limited number of case studies for validation and also faced potential bias in the selection of perspectives [16].

A study focused on the prioritization of software functionalities depending upon the developer's point of view. They utilized two different techniques for their study, i.e., spanning tree and AHP, for representing and ranking the software functions. The study highlighted limitations in terms of finding the reason why requirements were dependent on each other. Avoiding requirement dependencies may cause project delays. The study mentioned that the actual time needed for software development was longer than what was expected earlier, therefore emphasizing the need to decide priorities carefully for effective planning and management [17]. An approach was proposed with the objective of requirement selection and prioritization. Utilized rough set theory to prioritize requirements, which was applied to the educational institute examination system, and then compared with a fuzzy-based approach. Limited in terms of practical implementation challenges [6].

A new collaborative methodology has been introduced for the purpose of requirement prioritization, tackling two primary issues effectively: ambiguous priority and the relationships between requirements. For the purpose of accruing crucial information from the stakeholders, a survey matrix based on a questionnaire has been employed. The technique combines a variety of decision-making methods, including the intuitionistic fuzzy system, analytic hierarchy process, and interactive genetic algorithm, making the prioritization process all the more accurate. Future research work must focus on the further refinement of the methodology so that it may benefit the stakeholders and the developers in a better manner and encourage informed decision-making [18]. Further to the acceleration of the software system's functional requirements' prioritization, a graph-based method has been introduced for ranking requirements by using directed acyclic graphs. It helps assess the requirements' importance and relevance along with approximating the time the project will take for its completion. The research proposed that the preference of the stakeholders has a major impact on shaping the requirements' prioritization [19].

A methodology was outlined that involved software professionals in the requirements elicitation activity for software projects. The goal was to clarify their views about the impact of human factors, i.e., personality and motivation, on stakeholder collaboration, teamwork, decision-making activities, and the success of software projects. The data applied in this study are the results of a questionnaire survey conducted for software professionals involved in requirement engineering activity. This research paper does recognize some constraints, which also indicate the necessity for additional extensive studies in various fields. Further, it also highlights the necessity for providing practical suggestions for the purpose of guiding future work, adding greater robustness, and achieving higher generality of the results [20]. Another study postulated the idea of software requirements prioritization for working towards efficient software construction. This paper proposes the use case-based analytical hierarchical process (UC-Based-AHP) approach for software requirements prioritization, with the intention of improving AHP by reducing the pairwise comparisons with the implementation of NLP and available use cases [13].

In addition to this, a rough set theory was presented to prioritize software requirements. The study focused on the identification of potential stakeholders involved in the project, and requirements were ranked based on the decision-makers' opinion [21]. To overcome the challenges encountered in agile software development, a research study was proposed for the prioritization of non-functional requirements. Non-functional requirements were selected and analyzed to assess their effectiveness. However, this study faced a drawback, as customer or user involvement during the elicitation and prioritization of non-functional requirements is time-consuming and biased to some extent [1]. The SRPTackle approach was introduced to facilitate requirement prioritization through its semi-automated nature. Addressed the common issues faced in the existing prioritization method and utilized K-means and K-means++ grouping for the prioritization purpose. Future research needs to work on the approach's generalizability for maximizing effectiveness [22].

A hybridized ranking model was proposed, combining two methods, i.e., CBRanking and MACBETH, to prioritize the functional requirements of a software system under development. This hybridization increases stakeholder needs understanding and promotes informed decision-making, resulting in reliable prioritization findings. The study recommended applying the model to multi-stakeholder systems and in different software environments in the future [23]. In order to reduce work effort and cost, a study worked on the relationship between requirement similarity and software similarity. The idea was to retrieve code from existing software based on requirement relevance, hence minimizing resource utilization [24]. On a similar note, a method was presented to prioritize requirements depending on different decision criteria. Requirements were analyzed by a group of stakeholders based on the defined criteria [25].

The limitations and challenges faced in requirements engineering are documented. A survey questionnaire was employed to collect relevant data, which was analyzed using Expert Choice 11 software. The AHP was used to choose the success factors, which were the basis for the prioritization in decision-making. The findings determined that the user satisfaction factor greatly influences the success of requirement engineering. The authors recommend future studies involve a focus group so that the overall prioritization could be better understood [26]. An elaborate integrated framework was proposed to support the process of requirement prioritization. Fuzzy numbers and a 5-point scale for estimating the uncertainty and project stakeholders' subjective opinions were combined. The author advised the improvement of the framework by capturing and analyzing the real-world requirements in various development environments [27].

A semi-automated method for effective software requirements prioritization was put forward with the importance of efficiently executing software requirements prioritization. The suggested method catered to both quality and functional requirements by integrating automation with stakeholder input. They focused on improving the prioritization process and reducing time in manual work. The future work indicated the implementation of requirement tracing to manage changing requirements in an effective way [28]. Additionally, research was undertaken to assess the efficacy of methods adopted to prioritize non-functional requirements. The adopted method was AHP and the cost-value technique to prioritize. The author assesses the methods by implementing them in software projects with varying levels of complexities, i.e., low, medium, and hard. The research suggested the importance of both non-functional and functional requirements prioritizations in combination in order to obtain accurate results [29].

3 MATERIALS AND METHODS

Software success is dependent on stakeholder satisfaction. Multi-aspect requirement prioritization allows stakeholders to focus on specific requirements based on their relevance in the overall software project, considering different perspectives separately. This research framework is introduced for requirement prioritization using multi-aspect analysis in order to maximize the success rate of any software project. Framework working is illustrated in Figure 1.

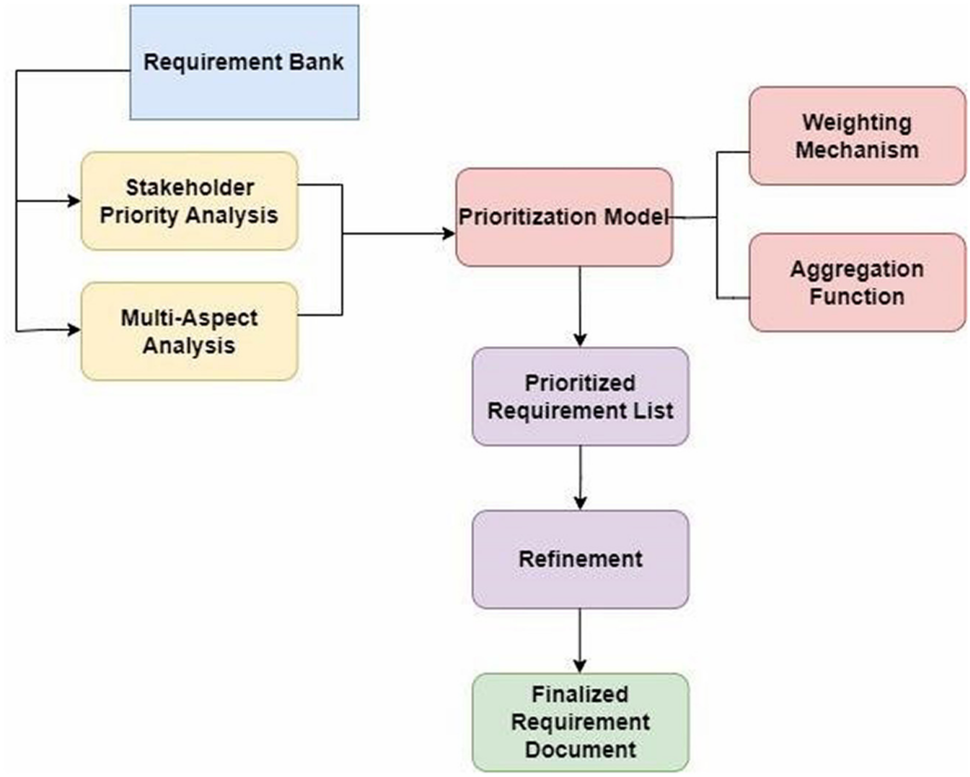

Fig. 1. Requirement prioritization framework

3.1 Requirement bank

The requirement bank constitutes all the requirements that need to be incorporated in a software project under development. The document under study must have all the functionalities and attributes defined for a developing system. This document contains all initial requirements provided by customers, business owners, designers, and developers working on the software project. Functional requirements (FRs) serve as a basic building block for any software product or software project. For this study, we need to prioritize these requirements based on multi-aspect analysis.

3.2 Stakeholder priority analysis

In this step of the framework, stakeholder priority analysis is performed. Customers, business owners, investors, designers, developers, and testers are involved in this stage, and they rate requirements based on their expert knowledge. They rate requirements based on the significant impact on the overall project. To get the stakeholder priorities list, we conducted a:

- Questionnaire
- Survey

3.3 Multi-aspect analysis

During the multi-aspect analysis phase, we highlighted the different aspects that are impacting the decision-making process during requirement prioritization. Multi-aspect analysis is the way of looking at a particular thing, considering different criteria, and taking multiple variables or elements into account. Opinions are assessed, doing sentiment analysis rather than just considering general feelings. The identified multi-aspects impacting the requirement prioritization process are listed below:

- Functional constraints
- Stakeholder preferences
- Requirement dependencies
- Business value

3.4 Prioritization model

A prioritization model will incorporate the multiple priorities that are defined and determined in the earlier stages by the software project stakeholders. Each requirement should be given a weight in the model that represents its significance to the project's stakeholders and overall quality. The prioritization model consists of the following two sections:

Weighting mechanism. In this step, the requirements obtained undergo stakeholder priority analysis and are then compared with the identified multi-aspects. Requirements are then assigned an appropriate score between 0 and 9 based on their significance in the overall software project. The scoring of requirements against each aspect is shown in Table 1.

Table 1. Requirement scoring against multi-aspect criteria

Software System Requirements	Functional Constraints (30%)	Stakeholder Preferences (35%)	Requirement Dependencies (20%)	Business Value (15%)
RE-I	5	9	4	6
RE-II	7	8	5	4
RE-III	6	6	5	3
RE-IV	8	9	4	2

Aggregation function. The aggregation function is applied in this stage in order to combine the weights assigned to each requirement in the earlier stage. The total score is calculated for each requirement, which represents the overall priority of requirements. To calculate the total weighted score, we use equations (1) and (2). The requirement for the overall weighted score and priority are shown in Table 2.

$$\text{Requirement Weight} = \text{Requirement score} * \text{Aspect weight} \quad (1)$$

$$\text{Total Weighted Score} = \text{Sum of specific requirement weights} \quad (2)$$

Table 2. Aggregated score table

Software System Requirements	Total Weighted Score	Requirement Priority
RE-I	6.35	3
RE-II	6.5	2
RE-III	5.35	4
RE-IV	6.65	1

3.5 Prioritized requirement list

The ultimate product of this study is the priority requirements list. In order to successfully develop a software product of high quality, the requirement priority is to be taken into account. The requirements are ranked in the order of the priority ratings given. The list serves as the deciding and planning tool of projects in which the most pressing needs receive the priority treatment.

3.6 Refinement

During this phase within the framework, after obtaining the list of requirements prioritized, one must go through cooperation and interaction with stakeholders involved in the project. This interaction is carried out with the aim of monitoring the impact of the prioritization process. The stakeholders go through the prioritized list of the requirements and provide recommendations in case there is the necessity to initiate the change to any parameter.

3.7 Finalized requirement document

This section contains the outcome of the framework in the form of a finalized requirement document. This document consists of the requirements with their defined priorities and dependencies, which are then followed by the designer and developer team.

4 RESULT AND DISCUSSION

The study highlighted the importance of requirement prioritization during the software development process in order to maximize the success rate by satisfying all stakeholder needs. The proposed framework reduces the risk of favorable requirements going unnoticed, thus avoiding suboptimal outcomes. The framework shall break down several factors that affect each specific requirement. A requirements multi-view analysis priorities framework offers much promise in terms of improving project achievements. The software development team shall successfully carry out the software components that undergo priority, which in turn improves the software engineering process's quality.

4.1 Overview of evaluation

The designed multi-aspect weighted prioritization framework aimed to assess four functional requirements—RE-I, RE-II, RE-III, and RE-IV—along with four evaluation aspects: Functional Constraints (30%), Stakeholder Preferences (35%), Requirement Dependencies (20%), and Business Value (15%). Each requirement had a continued assessment from 0 to 9 for every individual aspect, and the total weighted score (TWS) was calculated using the weighted sum model (WSM). This provided a basis for comparison of the requirements concerning their functional importance and the order in which they would be executed.

4.2 Aspect-wise contribution analysis

The contribution of each requirement to the total weighted score on each aspect, based on a stacked bar chart, is illustrated in Figure 2. For each requirement, it also illustrates how the constituent parts work together to generate the final total. It can be seen that Stakeholder Preferences and Functional Constraints drive all the requirements due to their greater weight (35 and 30%). RE-IV and RE-II in particular have the highest stacked segments of these components, explaining the higher totals. In contrast, RE-III appears to have less contribution from the high-weighted components, which suggests less priority for implementation. This visualization shows that the final outcome prioritization is driven by the balance of feasibility of implementation and stakeholder expectations.

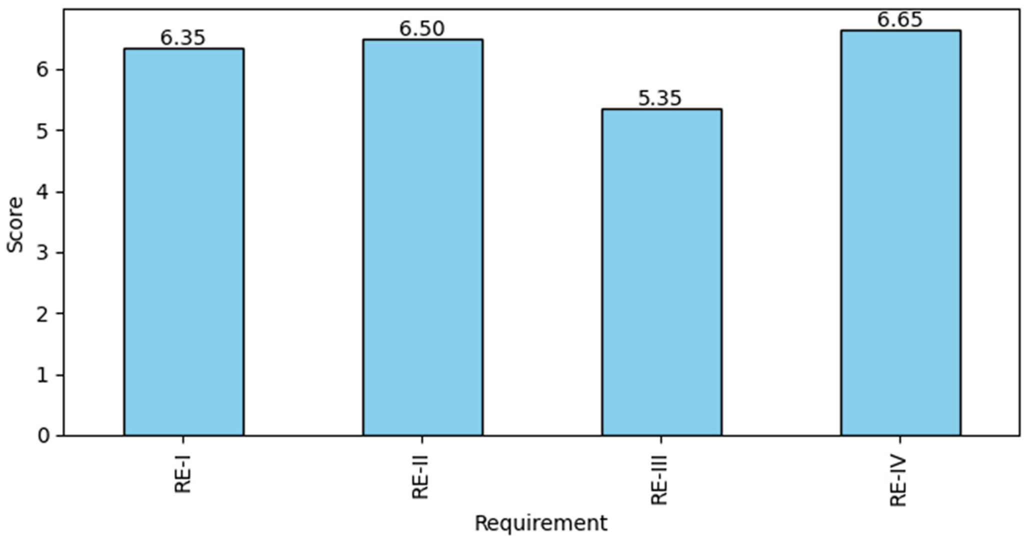

Fig. 2. Aspect-wise contribution of each requirement to the total weighted score

4.3 Requirement priority requirement

Figure 3 shows a bar chart of the TWS attributed to each requirement. The results clearly define the order of ranking with RE-IV (6.65) in the highest rank, followed by RE-II (6.50), RE-I (6.35), and RE-III (5.35). The distance between RE-IV and RE-III highlights a wide disparity in overall value. This indicates that implementing RE-IV with the highest priority would provide the maximum stakeholder satisfaction and business value with the given technical feasibility. The bar chart is an effective tool for displaying relative priority with a clear indication of the outstanding performance by RE-IV because of the full agreement between its scores for all the critical dimensions.

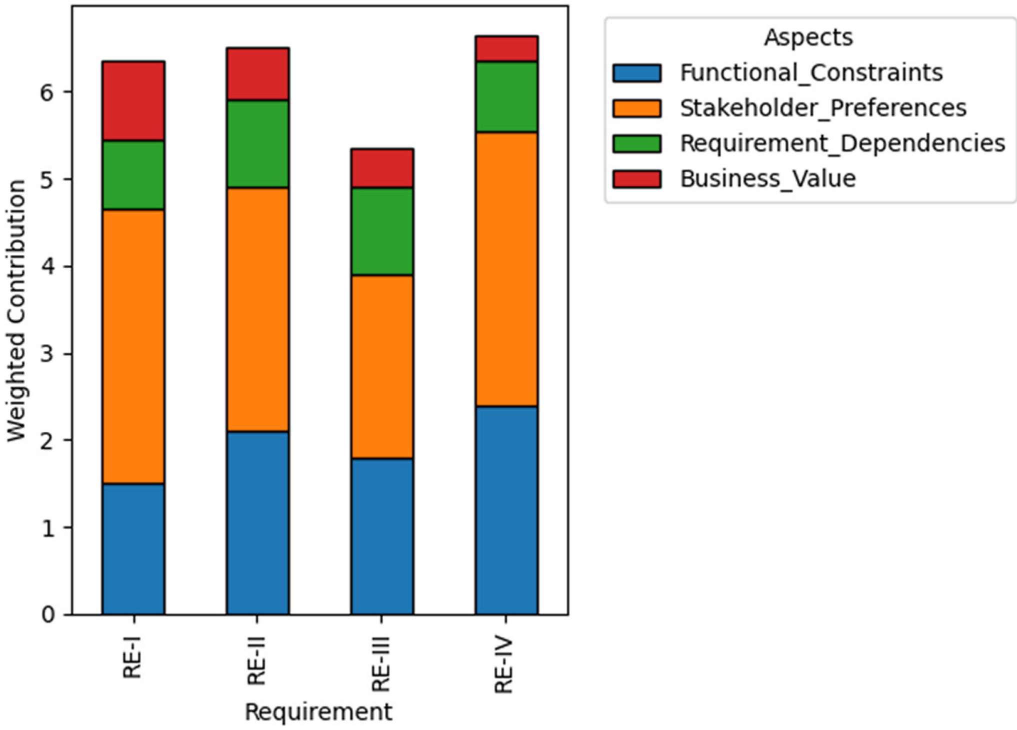

Fig. 3. Bar chart of total weighted score per requirement

4.4 Ranking trend visualization

In Figure 4, line graphs emphasize time series trends for total weighted score across all requirements. The slope of the graph from RE-IV to RE-III demonstrates a substantial decline, thereby visually supporting the ranking pattern. The slope of the line between RE-II and RE-I is nearly flat, meaning there is close competition in order of priority between the two requirements. Conversely, there is a substantial decline in slope from RE-II to RE-III, indicating that the requirements in that rank were significantly lower. This also illustrates the distribution of priorities for the framework and, most importantly, that the ranking pattern the proposed framework delivers is ranked logically and consistently without gaps or overlaps.

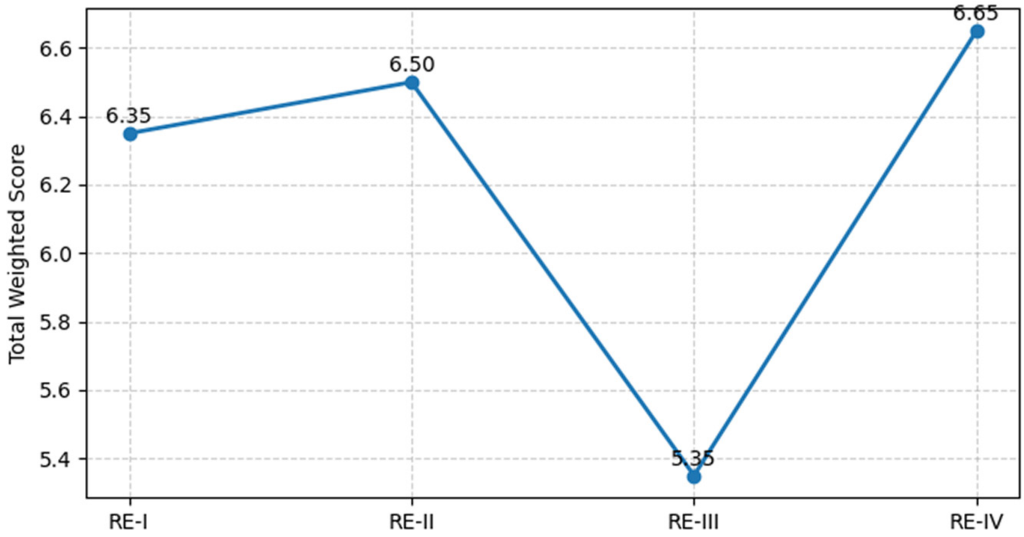

Fig. 4. Line plot showing ranking trend of requirements based on total weighted scores

4.5 Aspect-wise comparison across requirements

Figure 5 presents a grouped bar chart showcasing differences in the raw aspect scores across all the requirements. It highlights the most pronounced requirements in every aspect. For example, in the Functional Constraints and Stakeholder Preferences aspects, RE-IV scores the highest, with RE-II closely following in most of the other aspects. Conversely, RE-III has the most significant gaps concerning the Business Value and Stakeholder Preference aspects, which most adversely affect its overall weighted score. This grouped format underscores the requirements that excel in specific criteria complexities, which may help with deal prioritization and balancing trade-offs between conflicting stakeholders. Analyzing the subtotals adds clarity to the overall decision-making process.

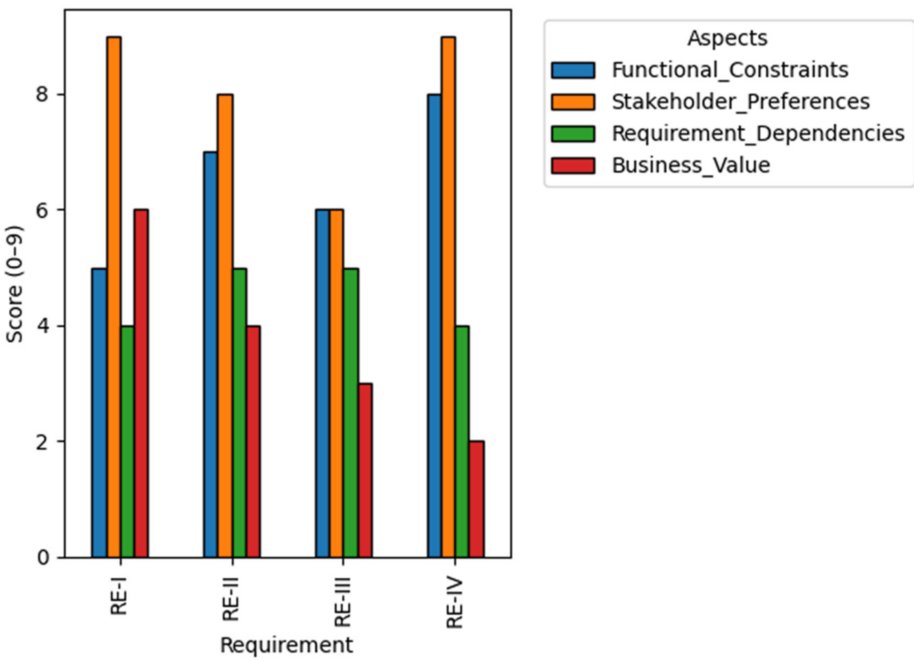

Fig. 5. Grouped bar chart comparing each aspect across requirements

4.6 Overall aspect influence

A summary of how much each element contributes to each requirement is displayed in a pie chart in Figure 6. Looking at the pie chart, it can be seen that Stakeholder Preferences occupy roughly 35% of the total weight, making them the strongest element within the prioritization framework. Following closely is Functional Constraints, with an almost equally impressive 30% weight, demonstrating the significance of the technical elements of practical implementation. The remaining 35% is split evenly between Requirement Dependencies and Business Value and shows moderate influence. The stakeholder and functional considerations dominate the priority of requirements, as the visualization shows, in line with the framework's weight distribution, confirming the logical arrangement of the framework.

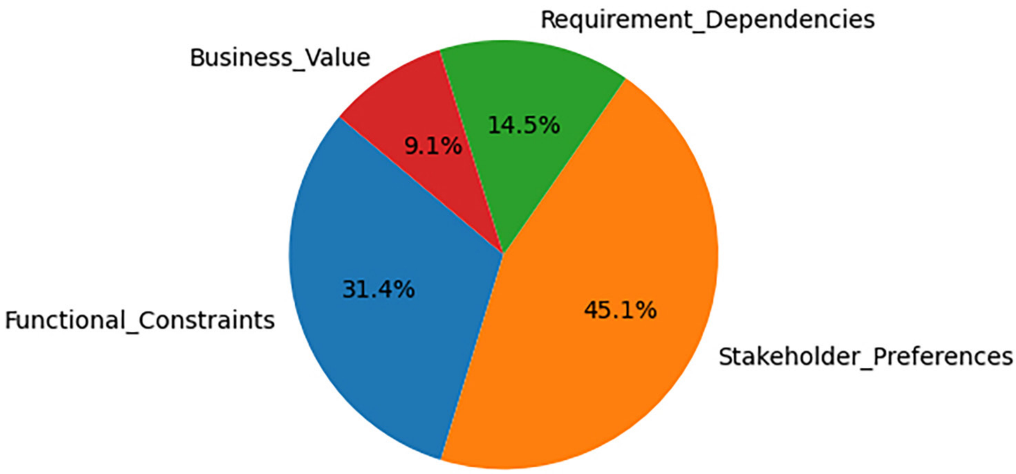

Fig. 6. Pie chart of overall aspect influence across all requirements

4.7 Discussion

As the consolidated findings from the previous figures demonstrate, the proposed multi-dimensional prioritization issue framework works. Technical feasibility, stakeholder importance, interdependencies, the business angle, and the scoring method unite within the scoring approach. The figures respond comprehensively to the interpretability needs of the teams rationally defending their chosen execution order, from detailed insights to overall rank trends. The approach pivoted unto the data, resolving subjectivity and guaranteeing fairness, consistency, and alignment to the organization and user priorities.

5 CONCLUSION AND FUTURE WORK

It presented a multi-dimensional weighted method of software requirement prioritization based on the four pillars of prioritization, such as Functional Constraints, Stakeholder Preferences, Requirement Dependencies, and Business Value. The method presented a scorecard that moved beyond qualitative opinions to simple prioritization results through the WSM. Results with empirical evidence showed that the highly prioritized requirement was RE-IV, which was motivated mostly by strong stakeholder and functional scores, followed by RE-II, RE-I, and RE-III. Graphical components revealed the weighted scores and rank information, which highlighted the effect of each component on the scores of requirements. The results showed explicit technical and stakeholder alignment gave decision-making around requirements more emphasis, clearness, and uniformity for potential integration and smooth execution.

Future studies can be conducted to enhance the proposed framework with optimization and machine learning models such that it could learn weights independently and be highly responsive to the fluctuating project requirements. AHP–TOPSIS, reinforcement learning, and fuzzy logic can be used to improve the accuracy and scalability in handling large collections of requirements. Stakeholder feedback in real time and dynamic adaptive recalibration methods will improve the dynamic response of the prioritization system to software requirements in an agile development environment. Further, human subjectivity in ratings may also be minimized by the text analysis of the requirements with the NLP techniques. The enhancements will move the system forward in the direction of an intelligent automated decision-support system for software requirements prioritization.

6 REFERENCES

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

[2] K. Ayub, F. Azam, M. W. Anwar, A. Amjad, and M. S. Jahan, "A novel approach for software requirement prioritization based upon non functional requirements," in *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 2019, pp. 8–15. https://doi.org/10.1109/CONISOFT.2019.00013

[3] F. A. Bukhsh, Z. A. Bukhsh, and M. Daneva, "A systematic literature review on requirement prioritization techniques and their empirical evaluation," *Computer Standards & Interfaces*, vol. 69, p. 103389, 2020. https://doi.org/10.1016/j.csi.2019.103389

[4] H. U. Nisa, S. U. R. Khan, S. Hussain, and W.-L. Wang, "An association rule mining-oriented approach for prioritizing functional requirements," *Computing*, vol. 106, pp. 2643–2674, 2024. https://doi.org/10.1007/s00607-024-01296-9

[5] H. Arshad, S. Shaheen, J. A. Khan, M. S. Anwar, K. Aurangzeb, and M. Alhussein, "A novel hybrid requirement's prioritization approach based on critical software project factors," *Cognition, Technology & Work*, vol. 25, no. 2, pp. 305–324, 2023. https://doi.org/10.1007/s10111-023-00729-3

[6] M. Sadiq and V. Susheela Devi, "Prioritization and selection of the software requirements using rough-set theory," *IETE Journal of Research*, vol. 69, no. 8, pp. 5169–5186, 2023. https://doi.org/10.1080/03772063.2021.1973593

[7] M. Ahmed, S. U. R. Khan, and K. A. Alam, "An NLP-based quality attributes extraction and prioritization framework in Agile-driven software development," *Automated Software Engineering*, vol. 30, no. 1, p. 7, 2023. https://doi.org/10.1007/s10515-022-00371-9

[8] Y. V. Singh, B. Kumar, and S. Chand, "Requirements prioritization using logarithmic fuzzy trapezoidal approach (LFTA)," in *Innovations in Information and Communication Technologies (IICT-2020)*, in Advances in Science, Technology & Innovation, P. K. Singh, Z. Polkowski, S. Tanwar, S. K. Pandey, G. Matei, and D. Pirvu, Eds., Cham: Springer, 2021, pp. 309–318. https://doi.org/10.1007/978-3-030-66218-9_36

[9] R. Devadas and N. G. Cholli, "Multi aspects based requirements prioritization for large scale software using deep neural lagrange multiplier," in *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, Villupuram, India, 2022, pp. 1–6. https://doi.org/10.1109/ICSTSN53084.2022.9761298

[10] A. Hudaib, R. Masadeh, M. H. Qasem, and A. Alzaqebah, "Requirements prioritization techniques comparison," *Modern Applied Science*, vol. 12, no. 2, pp. 62–80, 2018. https://doi.org/10.5539/mas.v12n2p62

[11] M. S. Jahan, F. Azam, M. W. Anwar, A. Amjad, and K. Ayub, "A novel approach for software requirement prioritization," in *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Mexico City, Mexico, 2019, pp. 1–7. https://doi.org/10.1109/CONISOFT.2019.00012

[12] A. E. Bakly and N. R. Darwish, "A proposed hybrid prioritization technique for software requirements based on fuzzy logic," *CiiT International Journal of Fuzzy Systems*, vol. 10, no. 2, pp. 45–52, 2018.

[13] M. Z. N. Maulana and D. Siahaan, "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)*, Yogyakarta, Indonesia, 2022, pp. 205–210. https://doi.org/10.1109/ICITISEE57756.2022.10057944

[14] T. Hassan, C. W. Mohammad, and M. Sadiq, "Using social network and fuzzy set theory for elicitation and prioritization of software requirements," *International Journal of Computer Theory and Engineering*, vol. 14, no. 3, pp. 126–134, 2022. https://doi.org/10.7763/IJCTE.2022.V14.1319

[15] R. B. Svensson and R. Torkar, "Not all requirements prioritization criteria are equal at all times: A quantitative analysis," *Journal of Systems and Software*, vol. 209, p. 111909, 2023. https://10.1016/j.jss.2023.111909

[16] I. Ibriwesh, S.-B. Ho, I. Chai, and C.-H. Tan, "Prioritizing solution-oriented software requirements using the multiple perspective prioritization technique algorithm," *Concurrent Engineering*, vol. 27, no. 1, pp. 68–79, 2019. https://doi.org/10.1177/1063293X18808559

[17] M. Yaseen, "Prioritization of software functional requirements from developers perspective," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 9, 2020. https://doi.org/10.14569/IJACSA.2020.0110925

[18] A. Gupta and C. Gupta, "A novel collaborative requirement prioritization approach to handle priority vagueness and inter-relationships," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 5, pp. 2288–2297, 2022. https://doi.org/10.1016/j.jksuci.2019.12.002

[19] M. Yaseen, A. Mustapha, S. Qureshi, A. Khan, and A. U. Rahman, "A graph based approach to prioritization of software functional requirements," *International Journal of Computer Science and Software Engineering*, vol. 9, no. 4, pp. 20–29, 2020. https://doi.org/10.47277/IJCSSE/9(4)1

[20] D. Hidellaarachchi, J. Grundy, R. Hoda, and I. Mueller, "The influence of human aspects on requirements engineering-related activities: Software practitioners' perspective," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 5, pp. 1–37, 2023. https://doi.org/10.1145/3546943

[21] M. Sadiq and V. S. Devi, "A rough-set based approach for the prioritization of software requirements," *International Journal of Information Technology*, vol. 14, no. 1, pp. 447–457, 2022. https://doi.org/10.1007/s41870-021-00749-0

[22] F. Hujainah, R. B. A Bakar, M. A. Abdulgabber, and K. Z. Zamli, "Software requirements prioritisation: A systematic literature review on significance, stakeholders, techniques and challenges," *IEEE Access*, vol. 6, pp. 71497–71523, 2018. https://doi.org/10.1109/ACCESS.2018.2881755

[23] I. P. Gambo, R. Ikono, O. G. Iroju, T. O. Omodunbi, and O. K. Zohoun, “Hybridized ranking model for prioritizing functional software requirements,” *International Journal of Software Innovation (IJSI)*, vol. 9, no. 4, pp. 19–49, 2021. https://doi.org/10.4018/IJSI.289167

[24] M. Abbas, A. Ferrari, A. Shatnawi, E. Enoiu, M. Saadatmand, and D. Sundmark, “On the relationship between similar requirements and similar software: A case study in the railway domain,” *Requirements Engineering*, vol. 28, pp. 23–47, 2023. https://doi.org/10.1007/s00766-021-00370-4

[25] G. D. Rottoli and C. Casanova, “Multi-criteria and multi-expert requirement prioritization using fuzzy linguistic labels,” *Paradigmplus*, vol. 3, no. 1, pp. 1–18, 2022. https://doi.org/10.55969/paradigmplus.v3n1a1

[26] I. Eitiveni and H. A. Hutapea, “Prioritizing critical success factors of requirements engineering using analytical hierarchy process,” *Jurnal Sistem Informasi*, vol. 19, no. 1, pp. 13–27, 2023. https://doi.org/10.21609/jsi.v19i1.1197

[27] I. Gambo, R. Ikono, P. Achimugu, and A. Soriyan, “An integrated framework for prioritizing software specifications in requirements engineering,” *International Journal of Software Engineering and Its Application*s, vol. 12, no. 1, pp. 33–46, 2018. https://doi.org/10.14257/ijseia.2018.12.1.03

[28] F.-F. Chua, T.-Y. Lim, B. Tajuddin, and A. P. Yanuarifiani, “Incorporating semi-automated approach for effective software requirements prioritization,” *Journal of Informatics and Web Engineering*, vol. 1, no. 1, pp. 1–15, 2018. https://doi.org/10.33093/jiwe.2022.1.1.1

[29] R. K. Chopra, V. Gupta, and D. S. Chauhan, “Experimentation on accuracy of non functional requirement prioritization approaches for different complexity projects,” *Perspectives in Science*, vol. 8, pp. 79–82, 2016. https://doi.org/10.1016/j.pisc.2016.04.001

7 AUTHORS

Zara Khalid is with the PMAS – Arid Agriculture University, University Institute of Information Technology, Rawalpindi, Pakistan (E-mail: zarakhalid571@gmail.com).

Inam Ullah is with the PMAS – Arid Agriculture University, University Institute of Information Technology, Rawalpindi, Pakistan.

Sidra Tahir is with the PMAS – Arid Agriculture University, University Institute of Information Technology, Rawalpindi, Pakistan.

Sadia Ali is with the PMAS – Arid Agriculture University, University Institute of Information Technology, Rawalpindi, Pakistan.

Ahtasham Sajid is with the Department of Computer Science, Fazaia Bilquis College of Education for Women’s PAF Nur Khan Base, Air University, Islamabad, Pakistan.

Sabitha Banu is with the Department of Computer Science & Cyber, PSGR Krishnammal College for Women, Coimbatore, Tamil Nadu, India.