Requirements Prioritization Techniques Review and Analysis

Raneem Qaddoura[1], Alaa Abu-Srhan[2], Mais Haj Qasem[3], Amjad Hudaib[3]
[1] Information Technology, Philadelphia University, Amman, Jordan
[2] Information Technology, Hashemite University, Amman, Jordan
[3] King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
E-mail: rqaddoura@philadelphia.edu.jo, alaaa@hu.jo, mais_hajqasem@hotmail.com, ahudaib@ju.edu.jo,

***Abstract*—Requirements prioritization is considered as one of the most important activities in the requirement engineering process. This paper gives an overview of the requirements prioritization activities and techniques. It also presents how data mining and machine learning techniques have been used to prioritize the software project requirements. A comparison between these techniques is also presented.**

Keywords— Requirements, Requirements Engineering, Requirements Prioritization Techniques

I. INTRODUCTION

Requirement engineering is considered at an early phase in the software development life cycle. The requirements engineering process includes several activities; feasibility study, requirements collection, classification, structuring, prioritization, validation, specification, and management [1]. Requirements prioritization (RP) is one of the most important activities in this process. Most software projects contain large number of requirements, these requirements need to be prioritized because resources are limited in budget and time and customer satisfaction is a main objective in software development. Therefore, stakeholders' involvement should be considered as they need to participate in prioritizing the requirements according to their importance so that requirements can be ordered in execution.

Since not all requirements can be executed at the same time, requirements are not naturally prioritized, and stakeholders have different points of view regarding the priority of each requirement, the requirements prioritization process should be taken in consideration where stakeholders should agree on the priority of requirements.

There are many techniques for requirements prioritization that are presented in literature, some techniques work well with projects having large number of requirements, other techniques are unable to give acceptable results when dealing with large number of requirements. In this paper, we present the most popular techniques used for requirements.

This paper is organized as follows: the next section introduces the requirements prioritization concept. Section 3 presents the most popular techniques. Section 4 provides comparison between the different techniques. The last section concludes the work.

II. REQUIREMENTS PRIORITIZATION

Requirements prioritization is an important task for decision makers. Requirements prioritization is essential in software engineering as it gives an optimal requirement implementation order for planning software releases and providing desired functionality as early as possible. With this regard, we can say that prioritizing requirements could mean prioritization by implementation order or prioritization by importance.

Implementing the most important functionalities first makes it possible to receive incremental feedback from the client, adjust schedule, solve errors, resolve misunderstandings between the organization and the client in early stages which leads to customer satisfaction. It can also be beneficial by removing unnecessary requirements that might be inefficiently costly, selecting the most appropriate requirements for each release, helping in future planning, lowering risk of cancellation, estimating benefits, prioritizing investments, and identifying financial impact for implementation of each requirement [2].

Requirements prioritization is critical and an important part of requirements analysis. Reasons behind this might be due to limitations in project resources where it is difficult to implement all requirements at the same time due to limitations in resources in terms of budget, staff, and schedule. Also, some projects might need several months or often multiple years to develop, so it is important to decide which requirement must be implemented first. Budget on the other hand plays an important factor when dealing with requirements prioritization process because budget for requirements engineering is often small compared to other activities in software engineering. Finally, Requirements have different level of importance and it is difficult to decide which requirement is the most important [2].

As discussed earlier, stakeholders' involvement should be considered in this process because different stakeholders have different perspectives and each one of them must decide what requirement has the highest priority in order to force stakeholders to openly address the relative importance of their requirements which leads to increased communication and consensus among stakeholders, provides a logical basis for requirement negotiation, and enables management and engineering to rationally schedule development activities.

978-1-5386-0527-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICTCS.2017.55

III. REQUIREMENTS PRIORITIZATION TECHNIQUES

Many techniques have been proposed for requirements prioritization which results in ordering requirements efficiently according to the importance in order to be considered in implementation. Many challenges are facing these techniques including budget, time, resources, technical constraints, the need for professional skills to implement these techniques, and the need to satisfy the clients' expectations. In addition, these techniques must work for all sizes of datasets including large and high volatile ones.

These techniques have some limitations regarding scalability, computational complexity, and/or reliability of results. Some lack full implementation, are prone to faults, or not valid in real cases. These techniques depend on experts in the field, need high communication with stakeholders, highly dependent on other requirements. This makes the job of proposing the right technique more difficult and makes improvements on these techniques highly needed.

Some reviews were made which analyzed and compared several requirements prioritization techniques. One of the recent reviews were done by Achimugu, Philip, et al. [3]. They identified and analyzed existing requirements prioritization techniques using search terms with relevant keywords and they classified primary studies under journal articles, conference papers, workshops, symposiums, book chapters and IEEE bulletins. Perini, A. et al. [4] presented an empirical study that evaluated AHP and CBRank requirements prioritization techniques in terms of ease of use, time-consumption, and accuracy. Aasem, M. et al. [5] examined several techniques and analyzed their applicability on software requirements domain and proposed a framework which uses some of the existing techniques. Khari, M. and Kumar, N. [6] conducted an experiment on some techniques and compared between them in terms of ease of use, total time taken, scalability, and accuracy. Wang, H et al. [7] compared between the prioritization matrix method and AHP method for HOQ applications in terms of practical applications in the industry. They concluded that prioritization matrix method is preferred when time, cost and difficulty are the major concerns in product improvement whereas AHP is preferred when accuracy is the major requirement.

Some other papers have discussed which requirements prioritization technique is best to prioritize software requirements, two of these papers are discussed here.

Karlsson et al. [8] found AHP to be the most capable technique amongst all the prioritization techniques as it provides the most efficient results which are on ratio scale, fault- tolerant and provides a consistency check. They carried out an experiment to evaluate six requirements prioritization techniques which are spanning tree, bubble sort, binary search tree, priority groups, AHP, and hierarchy AHP. They took 13 quality requirements for the experiment, and evaluated two types of measurements in their experiment; one for objective measure and the other for subjective measure. The experiment objective measures are: required number of decisions, total time consumption, and time consumption per decision, while the experiment subjective measures are: ease of use, reliability of results, and fault tolerance. They also discussed that amongst all the prioritization techniques, Bubble sort is another good technique that has lack of such important features. On the other side, AHP and bubble sort may be challenging for bigger problems as they have problem scaling up.

Hatton [9] conducted a case study to evaluate simple ranking, MoScoW, hundred dollars, and AHP requirements prioritization techniques. They used different criteria for evaluating these techniques which are ease of use, time to compete the overall prioritization process, and the stakeholder confidence. Their evaluation of the requirements prioritization techniques that was provided to the users are 12 requirements associated with cell phone features. A broad collection of human from different ages, sex, level of education and occupations were shortlisted as contestants. Also, they fulfilled 31 studies and the results were used for data analysis. An ordinal scale from 1 to 10 was used, to measure the difficulty and user confidence of the prioritization method.

We present below the most popular requirements prioritization techniques which are Analytic Hierarchy Process (AHP), Binary Search Tree (BST), Numeral Assignment Technique, Planning Game, Minimal Spanning Tree (MST), 100-dollar Method, Theory-W, Priority group, and Bubble sort.

A. Analytic Hierarchy Process (AHP)

AHP [10] is used for ordering requirements by decision makers, it uses a pair wise comparison matrix in order to compute the cost or value of requirements that are relative to each other. This technique ensures that the results are consistent and reliable.

AHP technique starts by creating an n x n comparison matrix where n is the number of requirements. n requirements are put in the rows and n requirements are put in the columns of the matrix as shown in table I. Then, for each unique pair of requirements, for example, Req1 and Req2, insert the value or the cost in the position where the row of Req1 meets the column of Req2. At the same time, the reciprocal values are inserted in the transposed positions (e.g. if cell Req1/Req2=3 then cell Req2/Req1=1/3). Finally, stakeholders discuss the comparison matrix and software managers prioritize the requirements and decide which will be executed and in what order it will be executed [11].

TABLE I. AHP COPARISON MATRIX

Year	Req1	Req2	Req3	Req4	Req5
Req1	1	3	1/2	5	1/6
Req2	1/3	1	1/2	1/5	1/5
Req3	2	2	1	1	4
Req4	1/5	5	1	1	¼
Req5	6	5	1/4	4	1

Many papers have analyzed and extended this technique. Parthasarathy, Sudhaman, and Maya Daneva [12] developed a new framework for ERP systems using AHP technique to prioritize ERP customization choices. Kamvysi, Konstantina,

et al [13] proposed and tested a framework for prioritizing students' requirements using Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) and the linear programming method (LP-GW-AHP). Govindan, Kannan, et al. [14] identified twenty-six barriers for a green supply chain management (Green SCM) and prioritized these barriers using AHP based on judgments of industrial experts.

B. Binary Search Tree (BST)

BST [15] suggested by HopcorftAho and Ullman, is used in sorting. There are two children at most of each node in a binary search tree and when using this tree for requirements prioritization, each requirement is represented in a node. The requirement is arranged on the tree according to its priority; the low priority requirements are put in the left side of the tree and high priority requirements are put on the right side.

BST technique starts by taking one requirement and put it on the root node. Then take the next requirement and compare it to the requirement on the root node. Depending on the comparison the requirement is put as a left child node if the requirement is less important than the root node and no left child node exists, it is put as the right child node if the requirement is more important than the root node and no right child node exists. If a left or right child node exist, then compare it to the left/right sub-tree. This is done until no requirements are left unassigned to the tree (see figure 1 as an example). Then, Requirements can also be put in a list (see figure 2 as an example), with the least important requirement at the bottom of the list and the most important requirement at the top of the list [10].

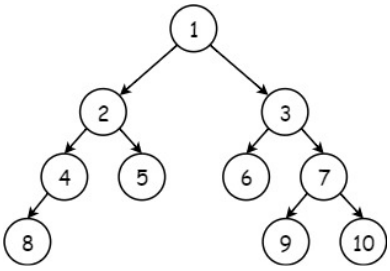

Fig. 1. BST Rquirement Tree.

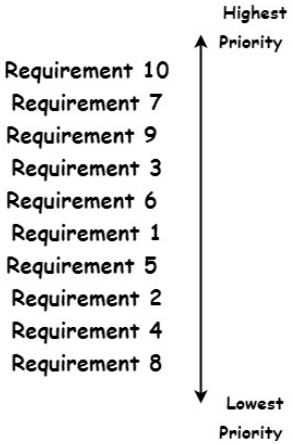

Fig. 2. BST Requirements List.

Some papers have analyzed and extended this technique. Bebensee T. et al. [16] used Binary Priority List for prioritizing requirements and compared between this technique and other techniques in terms of the process quality. Beg, Rizwan et al. [17] proposed a technique which uses the binary tree for requirements prioritization aiming at minimizing the number of comparisons in prioritizing requirement and keeping the process simple.

Some other studies have combined AHP with BST. Sadiq, Mohd, and S. K. Jain [18] proposed a technique for requirements prioritization using fuzzy based approach by combining a-level weighted f-preference relation and extented fuzzy-AHP in group decision making process; and then they used binary search tree sort technique to get the prioritized list of requirements.

C. Numerical Assignment Technique

This technique gives a scale for all requirements. This technique divides the requirements into three groups which are mandatory, desirable, and unessential [10]. we can assign each requirement with a number scale from 1 to 5 to identify its importance. First numbers meaning that it does not matter (the customer does not need it). Second, not important (the customer would accept its absence). Third, rather important (the customer would appreciate it). Fourth, very important (the customer does not want to be without it). Finally, mandatory (the customer cannot do without it). Where the final ranking is the average of all requirements rankings.

Some papers have been analyzing and extending this technique. Voolaa and Babu [19] studied how the imprecise data obtained from ENA can be aggregated using aggregation algorithms like; Multiple Attribute Utility Theory (MAUT), Interval Evidential Reasoning (IER), and Laplace Evidential Reasoning (LER) to generate requirements' priorities in the presence of conflicting personal preferences among assessors. They concluded that all the three algorithms are efficient in the sense that they provide rational and reliable results by providing the best possible solution for all the assessors, in the presence of competing personal preferences. However, LER has the potential to emerge as a competent aggregation algorithm when compared to MAUT and IER, because of its reasonable processing requirements when compared to IER and its ability to produce rich set of outputs when compared to MAUT.

Saini and Chomal [20] proposed a list of six parameters which were used to assign score to these prioritization techniques, and to indicated that which prioritization techniques is best suited for software projects carried out by post graduate courses in computer science. They concluded that Numerical Assessment Technique is hypothetical to be the finest technique taking academic context into consideration, due there it is easy to learn, execute and also consume less time for prioritizing requirements from student perspective.

D. Minimal Spanning Tree (MST)

MST method proposed by Karlsson [11]. The idea behind this method is that if the decision making is made absolutely constant, then the excess could be overwhelmed. For example,

if requirement R1 has greater priority than R2, and R2 has greater priority than R3, then we can conclude that R1 must have greater priority than R3.

E. Planning Game

The planning game [21] is one of the features of extreme programming (XP). This technique uses stories in order to prioritize the requirements. This technique considers as a variation of the Numeral Assignment Technique since the requirements also are classified into three groups: essential requirements in which the system will not function without it, less essential requirements which has a significant value, and not essential but it nice to have.

Karlsson et al. [22] described two consecutive controlled experiments comparing different requirements prioritization techniques with the objective of understanding differences in time-consumption, ease of use and accuracy. Their experiments evaluate Pair-wise comparisons and a variation of the Planning game and compare the Planning game to Tool-supported pair-wise comparisons. They concluded that the manual pair-wise comparisons is the most time-consuming of the techniques, and also the least easy to use, where Tool-supported pair-wise comparisons is the fastest technique and it is as easy to use as the Planning game.

F. Hundred Dollar Method

The hundred-dollar method [15] which is also called Cumulative Voting (CV) is a straight forward method. This technique gives the stakeholders of the system 100 imaginary units where this unit could be money, hours, etc. These units are used to distribute between the requirements where the requirement with higher units has a higher priority. For example, a stakeholder may give more units to a requirement and low units to another or distribute the units evenly between requirements.

This method has a drawback when there are too many requirements, then this method will not perform well and prioritization miscalculated, and the points do not add up to 100. It can also be difficult to keep track of how much has been assigned and what amount is left to dispose of.

Some papers have been analyzing and extending this technique. Rinkevic˘s and Torkar [23] proposed decision-making by collecting knowledge on the empirical use of CV which has many synonyms in literature: hundred (100) dollar ($) method/ test and hundred (100) point method, and develop a method for detecting prioritization items with equal priority by using a systematic literature review of CV and CV analysis methods. They also proposed Equality of Cumulative Votes (ECVs)—a CV result analysis method that identifies prioritization items with equal priority. They conclude that CV has been used in not only requirements prioritization and release planning but also in e.g. software process improvement, change impact analysis and model driven software development. Their evaluation of ECV indicates that it is able to detect prioritization items with equal priority and thus provide the practitioner with a more fine-grained analysis.

G. Theory-W

Theory-W which is also called "Win-Win" was developed in 1989 at the University of Southern California [10]. The idea behind this method is generating negotiations between stockholders in order to solve the disagreement about the project requirement between them where each of the stakeholders has a "win" requirement.

Each stakeholder must rank the requirements before the negotiations start, in this step the stakeholder must determine whether there are requirements that this stakeholder is willing to give up on, so the winning and losing conditions are fully understood.

Theory-W method is applied in four steps. Frist step separate the people from the problem. Second, focus on interests, not positions. Third invest options for mutual gain. Finally step, insist on using objective criteria.

H. Priority Groups

This technique described by Karlsson et al. [8]. This technique doesn't produce requirements group as an outcome, the final result is a ranked list of the requirements. The idea behind this technique is the same as numeral assignment technique, the difference between them is that numeral assignment technique groups the requirements only once but in priority Groups it groups the requirements repeatedly. Figure 3 shows how this technique groups the requirements into three group; low group, medium group, and high group.

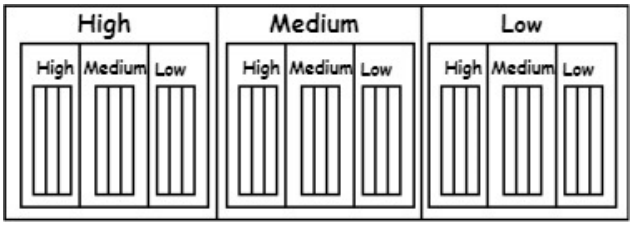

Fig. 3. Priority Groups technique [24].

There are five steps to applying the Priority Groups method. First step, gather all candidate requirements into one pile. Second, put each requirement into one of the three groups: high, medium or low priority. Third, in groups with more than one requirement, create three new subgroups (high, medium, low) and put the requirements within that group into these newly created subgroups. Fourth, repeat step 3 recursively until there is only one requirement in each subgroup. Finally, just read the requirements from left to right [24].

I. Bubble Sort

This method was described by Karlsson et al. [8]. The idea behind this technique is close to AHP. both of them utilize the pair-wise comparison activity, they need n x (n-1)/2 pair-wise comparisons, the difference between them that in bubble sort the decision maker only needs to determine which of the requirements among the two requirements is more important, not to what extent as in AHP.

Bubble Sort method is applied in five steps. First, outline the requirements in a vertical column. Second, compare the top two requirements from the column with each other to determine which is the most important. If the lower requirement is more important than the top one, swap their positions. Third, repeat this pair-wise comparison and swapping for the second and third requirement, then third and fourth requirement and so on until the bottom of the column is reached. Fourth, if any of the requirements have been moved during steps 2 and 3, repeat the process for the whole column starting again from the top two requirements (step 2). Finally, keep repeating this until no requirements are swapped during a complete pass through the column, which means that the requirements are now in priority order. The result will be a ranked column that continues requirements, the top of this column is the most important requirement, so the least important requirement is on the bottom of the column [24].

J. Machine learning and data mining techniques

Machine learning and data mining techniques were rarely found in the literature for prioritizing requirements.

Duan, C., Laurent, P., Cleland-Huang, J. et al. [25] proposed a method using data mining and machine learning techniques for requirements prioritization according to stakeholders' interests, business goals, and cross-cutting concerns such as security or performance requirements.

Avesani, P. et al. [26] proposed a case-based ranking framework for requirements prioritization which uses machine learning techniques to overcome the scalability problem. They compared their results with AHP in terms of expert elicitation effort and the requirements prioritization accuracy.

Perini, A., Susi, A., & Avesani, P. [27] described the Case-Based Ranking (CBRank) requirements prioritization method which combines project's stakeholders' preferences with requirements ordering approximations using machine learning techniques. They Concluded that their approach outperforms AHP with respect to the trade-off between expert elicitation effort and the requirements prioritization accuracy and that in the worst case, it works as well as AHP technique.

Tonella, P. et. Al. [28] proposed an Interactive Genetic Algorithm (IGA) for requirements prioritization and assessed the performance of the proposed algorithm in terms of effectiveness, efficiency, and robustness to decision maker errors. They concluded that the proposed algorithm outperforms Incomplete Analytic Hierarchy Process (IAHP).

Pitangueira, A. M. et al. [29] presented a review on Search-Based Software Engineering (SBSE) approaches for requirement selection and prioritization problems and provided quantitative and qualitative assessment for these approaches.

Wang, X., & Zeng, H. [30] proposed an approach for Test Case Prioritization for Requirement Properties based on historical data. Experimental results showed an improved performance using measurements of Average Percentage of Faults Detected and Fault Detection Rate.

IV. COMPARISON BETWEEN REQUIREMENTS PRIORITIZATION TECHNIQUES

Based on the discussion given above for the first 9 techniques, we compare these techniques in terms of complexity, ease of use, reliability of results, fault tolerance, speed, and number of requirement as shown in table II. The level of evaluation criteria for each technique are interpreted as high, medium, or low.

Numerical Assignment, planning game, and 100 dollar techniques have the lowest complexity as they have minimum computations compared to the other techniques. Numeral Assignment Technique and 100 dollar techniques have high ease of use, speed, and number of requirements whereas planning game and 100 dollar techniques have higher fault tolerance than Numerical Assignment Technique.

BST, MST, and Priority group have high reliability, high fault tolerance, and are easy to use and they are all have medium speed, and number of requirements but priority group and BST have higher complexity than BST because the idea of MST method is that if the decisions are made perfectly consistent, the redundancy will not exist, and in this case the number of comparisons will reduce to only n-1 comparisons.

AHP has low reliability, fault tolerance, and speed compared to the other techniques and can handle low number of requirements as it has relatively high computational complexity.

TABLE II. COMPARISON OF REQUIREMENTS PRIORITIZATION TECHNIQUES

Evaluation Criteria	AHP	BST	Numeral Assignment Technique	MST	Planning game	Hundred Dollar Method	Theory-W	Priority Groups	Bubble Sort
Complexity	High	Meduim	Low	Low	Low	Low	Low	Meduim	High
Ease of Use	Medium	High	High	High	Medium	High	High	High	Low
Reliability of Results	Low	High	Low	High	High	Medium	Medium	High	Low
Fault tolerance	Low	High	Low	High	Medium	Medium	Medium	High	Low
Speed	Low	Medium	High	Medium	Low	High	Low	Medium	Low
Number of Requirements	Low	Medium	High	Medium	Medium	High	Low	Medium	Low

Bubble sort has the highest complexity and low speed. It can handle low number of requirements compared to the other techniques but it is easy to use, reliable, and has low fault tolerance.

Assignment Technique and 100 dollar can handle high number of requirements due to their low computational complexity whereas AHP, Theory-W, and bubble sort relatively cannot handle high number of requirements due to their relatively high computational complexity.

I. Conclusion and Future Work

This paper presented the most popular techniques for requirements prioritization and their related literature. Machine learning and data mining techniques are also presented. There are many techniques used for requirements prioritization and because a software engineer should select one of those techniques, considerations must be taken regarding the number of requirements for the project and the type of the project in terms of reliability, speed, fault tolerance, and time consumption. For this reason, we cannot say that a technique is considered as the best one for all types of projects. In the future, we intend to investigate more about machine learning and data mining techniques and compare between these techniques against the already discussed techniques.

References

[1] Sommerville, I., & Sawyer, P. (1997). Requirements engineering: a good practice guide. John Wiley & Sons, Inc..

[2] Ibrahim, O., & Nosseir, A. (2016). A Combined AHP and Source of Power Schemes for Prioritising Requirements Applied on a Human Resources. In MATEC Web of Conferences (Vol. 76, p. 04016). EDP Sciences.

[3] Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. R. (2014). A systematic literature review of software requirements prioritization research. Information and software technology, 56(6), 568-585.

[4] Perini, A., Ricca, F., & Susi, A. (2009). Tool-supported requirements prioritization: Comparing the AHP and CBRank methods. Information and Software Technology, 51(6), 1021-1032.

[5] Aasem, M., Ramzan, M., & Jaffar, A. (2010, June). Analysis and optimization of software requirements prioritization techniques. In Information and Emerging Technologies (ICIET), 2010 International Conference on (pp. 1-6). IEEE.

[6] Khari, M., & Kumar, N. (2013). Comparison of six prioritization techniques for software requirements. Journal of Global Research in Computer Science, 4(1), 38-43.

[7] Wang, H., Xie, M., & Goh, T. N. (1998). A comparative study of the prioritization matrix method and the analytic hierarchy process technique in quality function deployment. Total Quality Management, 9(6), 421-430.

[8] Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. Information and software technology, 39(14-15), 939-947.

[9] Hatton, S. (2007, November). Early prioritisation of goals. In International Conference on Conceptual Modeling (pp. 235-244). Springer Berlin Heidelberg.

[10] Mead, N. R. (2006). Requirements prioritization introduction. Software Eng. Inst. web pub., Carnegie Mellon Univ.

[11] Karlsson, J. (1996, April). Software requirements prioritizing. In Requirements Engineering, 1996., Proceedings of the Second International Conference on (pp. 110-116). IEEE.

[12] Parthasarathy, S., & Daneva, M. (2014). Customer requirements based ERP customization using AHP technique. Business process management journal, 20(5), 730-751.

[13] Kamvysi, K., Gotzamani, K., Andronikidis, A., & Georgiou, A. C. (2014). Capturing and prioritizing students' requirements for course design by embedding Fuzzy-AHP and linear programming in QFD. European Journal of Operational Research, 237(3), 1083-1094.

[14] Govindan, K., Kaliyan, M., Kannan, D., & Haq, A. N. (2014). Barriers analysis for green supply chain management implementation in Indian industries using analytic hierarchy process. International Journal of Production Economics, 147, 555-568.

[15] Khan, J. A., Rehman, I. U., Khan, Y. H., Khan, I. J., & Rashid, S. (2015). Comparison of requirement prioritization techniques to find best prioritization technique. International Journal of Modern Education and Computer Science, 7(11), 53.

[16] Bebensee, T., van de Weerd, I., & Brinkkemper, S. (2010, June). Binary priority list for prioritizing software requirements. In International working conference on requirements engineering: foundation for software quality (pp. 67-78). Springer Berlin Heidelberg.

[17] Beg, R., Abbas, Q., & Verma, R. P. (2008, July). An approach for requirement prioritization using b-tree. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp. 1216-1221). IEEE.

[18] Sadiq, M., & Jain, S. K. (2014). Applying fuzzy preference relation for requirements prioritization in goal oriented requirements elicitation process. International Journal of System Assurance Engineering and Management, 5(4), 711-723.

[19] Voola, P. (2017). Study of aggregation algorithms for aggregating imprecise software requirements' priorities. European Journal of Operational Research, 259(3), 1191-1199.

[20] CHOMAL, V. S. A Parameters-based Approach for Requirement Prioritization for Software Development in Academic Context.

[21] Beck, Kent, and Cynthia Andres. "Extreme Programming Explained: Embrace Change." Reading: Addison-Wesley Professional (2004).

[22] Karlsson, L., Thelin, T., Regnell, B., Berander, P., & Wohlin, C. (2007). Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritisation techniques. Empirical Software Engineering, 12(1), 3-33.

[23] Riņķevičs, K., & Torkar, R. (2013). Equality in cumulative voting: A systematic review with an improvement proposal. Information and Software Technology, 55(2), 267-287.

[24] Vestola, M. (2010). A comparison of nine basic techniques for requirements prioritization. Helsinki University of Technology.

[25] Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. Requirements engineering, 14(2), 73-89.

[26] Avesani, P., Bazzanella, C., Perini, A., & Susi, A. (2005, August). Facing scalability issues in requirements prioritization with machine learning techniques. In Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on (pp. 297-305). IEEE.

[27] Perini, A., Susi, A., & Avesani, P. (2013). A machine learning approach to software requirements prioritization. IEEE Transactions on Software Engineering, 39(4), 445-461.

[28] Tonella, P., Susi, A., & Palma, F. (2013). Interactive requirements prioritization using a genetic algorithm. Information and software technology, 55(1), 173-187.

[29] Pitangueira, A. M., Maciel, R. S. P., & Barros, M. (2015). Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. Journal of Systems and Software, 103, 267-280.

[30] Wang, X., & Zeng, H. (2016, May). History-based dynamic test case prioritization for requirement properties in regression testing. In Continuous Software Evolution and Delivery (CSED), IEEE/ACM International Workshop on (pp. 41-47). IEEE.