

# Using Binary Priority List and Priority Groups to Prioritize Functional Requirements: ODOO ERP as Case Study

Muhammad Yaseen<sup>1</sup>, Aida Mustapha<sup>2</sup>, Atta Ur Rahman<sup>3</sup>, Shams Uz Zaman<sup>4</sup>, Syed Wajid Kamal<sup>5</sup>

<sup>1,2</sup> Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia

<sup>3</sup> Department of Software Engineering, University of Swat

<sup>4,5</sup> Department of Computer Science and Software Technology Abasyn University Peshawar

yaseen\_cse11@yahoo.com

**Abstract - Requirements prioritization is an important activity during requirements collection and management phase of software engineering. Prioritization of small size requirements is not an issue but its importance increases when requirements are large in size. Different prioritization techniques are suggested to prioritize different types of requirements but none of the suggested techniques are applied to prioritize software requirements based on dependencies in functional requirements (FRs) from the perspective of software developers. This study addresses this research gap by applying Cumulative Voting (CV) and Priority Groups based prioritization approach to prioritize FRs. Using spanning trees, all FRs are first inter-related and then CV is applied first to assign priority values to each requirement belong to different spanning trees. Using priority groups, prioritized requirements belong to different spanning trees are assigned to priority groups such that all FRs inside each priority groups have similar priority. To do so, FRs of ODOO ERP as case study are considered for prioritization. Prioritized FRs can be easily managed by parallel development team members without having dependencies in FRs of priority group.**

## 1. INTRODUCTION

Requirement prioritization (RP) is an important activity during requirements management and implementation phase in requirement engineering [1][2]. RP is about giving priority to requirements for better time planning during implementation [3][4]. Without requirements prioritization, development phase could be costly as it takes more efforts to complete all the requirements within the specified timeframe [5]. Existing techniques from the literature have shown that they are not scalable for large set of requirements especially in dealing with dependency issues between the functional requirements. There are three types of requirements; (1) business requirements which deals with benefits and cost issues of requirements along with time constraint, (2) functional requirements (FRs) which are necessary for software system to develop, and (3) non-functional requirements that are not directly demanded but are necessary for ensuring quality product such as security and performance issues. Along this line, research on requirements prioritization techniques depends on the type of requirement under study whether at business level [6][7], non-functional level [8] or at functional level [9]. Meanwhile, some of the techniques are able to cater for all types of requirements [10]. In addition to the types of requirements, requirements prioritization techniques also depends on the size of the requirements data. For example, AHP is suitable for a small set of functional requirements but often fail on large requirements due to high time consumption. Technique such as cumulative voting (CV) works well for medium size requirements and easy in use. A lot of work is done and dozens of prioritization techniques are presented but none of them are applied based on inter-dependencies in FRs. As a case study, FRs of ODOO ERP are considered for prioritization. In our previous research studies, FRs of ODOO ERP are considered [11][12].

## 2. LITERATURE REVIEW

Analytical Hierarchical Process (AHP) is an organized decision-making method that is intended to compute complex multi-criteria decision problems [17]. AHP is technique that is also applied efficiently in many other fields such as biology and social sciences for prioritization. In fact, AHP is the utmost frequently discussed prioritization technique within decision making in requirements engineering. AHP is led by comparing all possible pairs of hierarchically categorized entities such as requirements as well as stakeholders for obtaining comparative priorities for all objects. For each pair to compare, the prioritizing person estimates the importance relationship between the objects on a nine-level scale, where 1 means equal importance and 9 is the maximum difference. If requirement  $i$  is assigned an importance value when compared to requirement  $j$ , then  $j$  has the reciprocal value when compared to  $i$ . This shows that each pair of requirements only needs to be compared once with total of  $n(n-1)/2$  (where  $n$  is the number of requirements) comparisons. Cumulative voting (CV) is a method where stakeholders are given 100 dollars and they have to distribute on all possible requirements just like voting mechanism [10]. This technique is also known as 100 dollars method. The requirement with high votes will be

given more priority. This method is very simple for assigning priority to requirements but disadvantage of using this technique is that votes distribution of large size requirements become difficult. Besides, it stakeholders have to assign score to requirements manually. Priority Groups or Numerical Assignment is technique suggested where instead of prioritizing individual requirements, requirements are assigned to different priority groups and groups are then assigned priority values. This technique is suitable for large size requirements [13]. Prioritization techniques such as Goal-based Prioritization [7] is suggested to prioritize user requirements based on goals and business requirements. In Value Based technique, requirements are prioritized on the basis of weights of core values of business [14]. Company stakeholders or managers use ordinal scale to score business values that are necessary for organization. For example for banking system, the value of security is very high as compare to other values [15]. Prioritizing requirements on the basis of its cost and how much it has advantages is necessary. Techniques such as benefit and cost prediction [16] is suggested to prioritize requirements on the basis of cost and benefits of particular requirements for their business. In his research study, author [17] describe prioritization based on cost of the requirements and benefits it provide to customers. Case-Based Ranking (CBR) a machine learning approach is presented to reduce the efforts during prioritization which combines stakeholder preferences with requirement ordering approximations computed through machine learning approaches. The framework provides an iterative prioritization process that can handle single and multiple human decision makers (stakeholders) and different ordering criteria [18].

### 3. RESEARCH METHOD DESIGN

Design of research method consist of the following steps as shown in Figure 1. In first step, all FRs are inter-related using Directed Acyclic Graph (DAG) and then converted to possible number of spanning trees as discussed in section 3.1 below. In second step, all FRs belong to different spanning trees are prioritized using CV technique as discussed in section 3.1. In third step, using NA, all prioritized requirements are assigned to possible number of priority groups. In last step, results are analyzed using number of dependencies in FRs inside each priority group. Reducing number of dependencies in FRs inside each priority group shows importance of prioritization. All steps of prioritization are explained below.

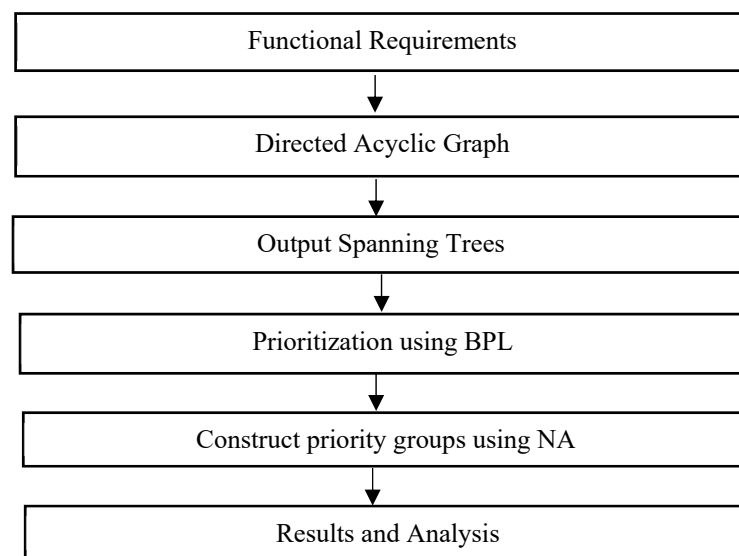


Figure 1: Step by step research design process

#### 3.1. Output Spanning Trees

A graph  $G = (V;E)$  consists of a finite set of vertices  $V$  and a finite set of edges  $E$ . Graphs are useful for the representation of any kind of data in particular sequence. This research uses directed acyclic graphs (DAG) rather than cyclic graphs. A directed graph  $E$  is a set of ordered pairs of vertices  $(u; v)$ . The arrows in the graph indicates the dependency of a requirement on another requirement. The requirement generates arrow and points to another requirement indicating that it is necessary or required for another requirement. For example,  $R1 \leftarrow R2$  indicates that  $R1$  is depended on  $R2$  or  $R2$  is required for the completion of  $R1$ . Figure 2 shows the graph representation of requirements through directed acyclic graph.

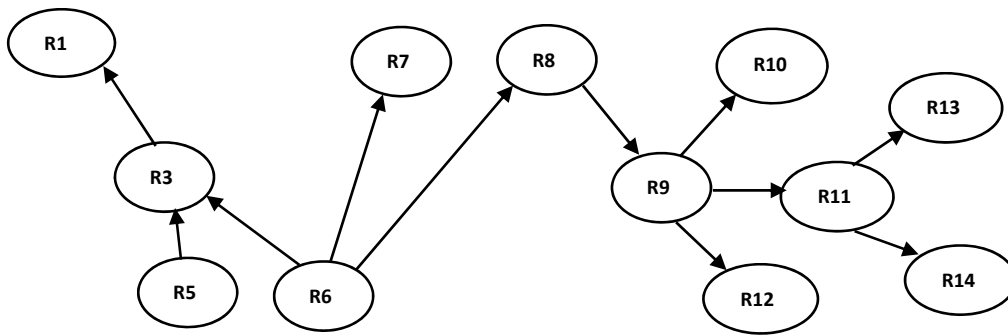


Figure 2: Directed graph inter-relating FRs

Resulted spanning trees from DAG of Figure 2 are shown in Figure 3 below. Spanning trees are produced from directed graph easily by following depth first search (DFS) method as discussed in [11].

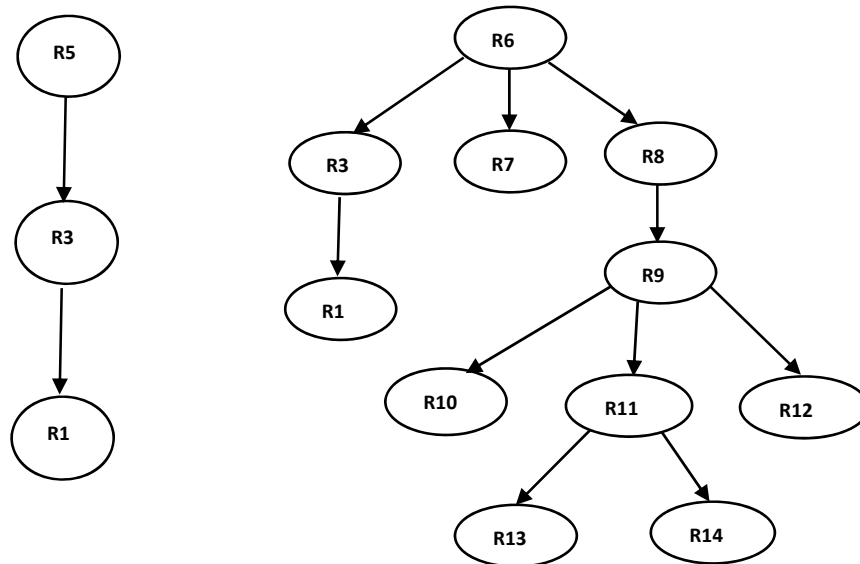


Figure 3: Resulted spanning trees from Figure 2

**3.2. Prioritization using Cumulative Voting**

Requirement that is required for other requirement will get high votes e.g. In Figure 3, R6 will get high priority votes as compare to R3, R7 and R8 while R8 will get high priority votes as compare to R9. Similarly, R9 will get high priority votes as compare to R10, R11 and R12. In this way, R11 will get high priority votes as compare to R13 and R14. As an example, priority votes assign to each requirement are given in Table 1 of both trees. Sum of priority values as result of prioritization of both trees will be equal to 100.

Table 1: Prioritized FRs of two spanning trees of Figure 3

Requirement	Votes	Requirement	Votes
R6	L1	R11	L4
R3	L2	R12	L4
R7	L2	R13	L5
R8	L2	R14	L5
R1	L3	R5	L1
R9	L3	R3	L2
R10	L4	R1	L3

### 3.3. Priority Groups

Next, the prioritized list of FRs will be assigned to different priority groups. This approach assigns requirements into priority groups, whereby instead of prioritizing individual FRs, groups are prioritized such that high priority FRs belong to different trees with same priority will be assigned to high priority group and low priority FRs will be assigned into low priority group. Low priority group FRs cannot be implemented until high priority group FRs are not implemented. E.g. Figure 4 shows order of priority groups while requirements are assigned in order of priority votes in descending order. When number of requirements is large, it is difficult to prioritize especially when software requirements are to be implemented by parallel development. If two depended requirements are assigned to parallel developers, this can cause problem as one requirement can only be implemented when its pre-requisite requirement is implemented first. Priority groups help in reduction of dependencies in requirements of parallel developers as inside priority group, all requirements are same or near to same in priority and if requirements are near to same in priority shows these requirements are not depended because depended requirements are assigned different votes during prioritization.

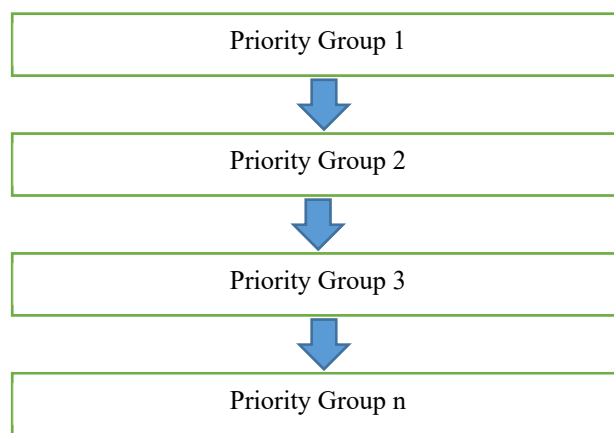


Figure 4: Requirements assigned to priority groups

### 4. CASE STUDY

Table 2 shows FRs of ODOO ERP. Column requirement shows all FRs belong to different modules of ODOO ERP while column required for shows all requirements for each particular requirement is required. Table 2 shows requirements belong to different spanning trees from Table. Requirements belong to each spanning tree are categorized into different levels and requirements of each level will be assigned same priority points.

Table 2: FRs belong to different modules of ODOO ERP

Requirement	Required for	Requirements	Required for	Requirements	Required for
R1 (Employee)	R2,R4,R10, R11,R12, R17,R18, R20, R21, R22, R23, R25, R67, R81,	R33 (customer detail)	R24, R35, R36, R39,R55, R61, R64, R73, R90, R55	R65 (supplier ledgers)	
R2 (Public information's of employee)		R34 (products detail)	R35, R42, R60, R66,R70, R71, R91, R61	R66 (stock ledgers)	
R3 (Employee personal info)		R35 (sale)	R32, R51, R61, R62,	R67 (HR expense management)	
R4 (Contact info)		R36 (customer refund)		R68 (purchase return view)	

R5 (Job position)		R37 (Sales persons)	R35, R36, R58, R63,	R69 (sale return view)	
R6 (Department)	R5, R81, R67	R38 (customer receipts)		R70 (Transfer In)	
R7 (Job information's)		R39 (customer payment)	R38, R55	R71 (Transfer out)	
R8 (Manager)	R5, R24, R67	R40 (supplier receipts)		R72 (order to suppliers)	
R9 (Coach)		R41 (supplier detail)	R42, R44, R52, R60, R65, R72	R73 (order from customer)	
R10 (Contract information's)		R42 (purchase)	R51, R59	R74 ()	
R11 (Contract reference information's )		R43 (Sales man)	R42, R44	R75 (Balance sheet)	
R12 (Salary generation)	R21,	R44 (supplier refund)		R76 (compose message)	R79
R13 (Salary rules)		R45 (supplier payment)	R40,	R77 (message inbox)	R80
R14 (Salary structure)	R12	R46 (bank statement)	R47	R78 (message Draft)	
R15 (Salary categories)	R12	R47 (bank detail)	R49, R50, R53	R79 (sent messages)	
R16 (Registers)	R12, R13,	R48 (cash registers)		R80 (message Searching)	
R17 (Apply for leave)	R19,R20,	R49 (put money in)		R81 (Job position in recruitment)	
R18 (Allocation request)		R50 (put money out)		R82 (Job)	
R19 (Approval)		R51 (Profit and lost)		R83 (appraisal form)	
R20 (Leave summary)		R52 (supplier payment)		R84 (create a job position)	
R21 (HR payroll)		R53 (Journals accounts)	R54	R85 (Recruitment form)	
R22 (HR Expenses)		R54 (Chart of accounts)		R86 (Job selection process)	
R23 (HR expenses)		R55 (Analytic accounts)	R54	R87 (Link tracker)	
R24 (Project management)	R26, R27, R28, R29	R56 (company)		R88 (Mass mailing)	
R25 (Add team members)		R57 (region)	R58	R89 (contacts)	
R26 (Extra information's)		R58 (Area)		R90 (business pipeline)	
R27 (Project stages)		R59 (purchase view)		R91 (manufacturing orders)	
R28 (View current task)		R60 (purchase return)	R68,	R92 (fleet management)	R93,

R29 (create a task)	R31,	R61 (sale return)	R69	R93 (Vehicle repairing)	
R30 (Extra information's)		R62 (sale view)		R94 (Directories for documents)	R96
R31 (Tasks stages)		R63 (salesman ledgers)		R95 (Documents history)	R96
R32 (customer invoice)	R36	R64 (customer ledgers)		R96 (Documents attachments)	

Table 3 shows FRs belongs to different spanning trees. Column Tree shows all notations for particular spanning trees. Requirements in each spanning tree belong to different level e.g. in spanning tree T3, R8 is required for R5, R67 and R24. So, root node R8 will be kept in Level 1 category and R24, R5 and R67 will be kept in Level 2 category. Similarly, R24 is required for R26, R27, R28 and R29, so these requirements will be kept in Level 3 category. R29 is required for R31, so it will goes to Level 4. Purpose of keeping requirements in different levels is distribute votes equally among them as these requirements are not depended on each other's. Also vote's distribution process will become easier. While 20 requirements are not inter-related to any other requirements.

Table 3: FRs belong categorized to different levels in spanning trees

Tree	Level 1	Level 2	Level 3	Level 4
T1	R1	R81, R25, R2, R23, R67, R4, R10, R11, R12, R17, R18, R22	R19, R20, R21	
T2	R6	R5, R67, R81		
T3	R8	R5, R67, R24	R26, R27, R28, R29	R31
T4	R34	R42, R60, R66, R35, R70, R71, R90	R51, R59, R68, R32, R61, R62, R80	R36, R69
T5	R43	R42, R44	R51, R59	
T6	R41	R44, R65, R72, R42, R52, R60	R51, R59, R68	
T7	R37	R58, R63, R35	R32, R61, R62	R36, R69
T8	R46	R47	R49, R50	
T9	R33	R73, R35, R64, R39	R62, R61, R32, R69, R38	R36, R69
T10	R16	R12, R13	R21	
T11	R54	R53, R55		
T12	R57	R58		
T13	R14	R21		
T14	R15	R21		
T15	R76	R79		
T16	R45	R40		
T17	R91	R92		
T18	R93	R94	R95	R96

### 5. RESULTS AND ANALYSIS

Results of prioritization as result of prioritization are shown in Table 4 below in descending order or priority. Any number of priority groups can be constructed using NA. Top priority requirements will be assigned to high priority group will low priority requirements will be assigned to low priority groups.

Table 4: Prioritized FRs of all spanning trees as result of BPL

Requirement	Priority	Requirement	Adjusted Priority
R1	L1	R90	L2
R14	L1	R92	L2
R15	L1	R94	L2
R16	L1	R10	L2
R33	L1	R11	L2
R34	L1	R12	L2
R37	L1	R13	L2
R41	L1	R100	L2
R43	L1	R68	L3
R45	L1	R49	L3
R46	L1	R50	L3
R54	L1	R51	L3
R93	L1	R59	L3
R57	L1	R80	L3
R6	L1	R61	L3
R8	L1	R62	L3
R99	L1	R95	L3
R42	L2	R38	L3
R47	L2	R19	L3
R44	L2	R20	L3
R17	L2	R21	L3
R18	L2	R32	L3
R2	L2	R26	L3
R4	L2	R27	L3
R5	L2	R28	L3
R22	L2	R29	L3
R23	L2	R69	L4
R24	L2	R36	L4
R25	L2	R96	L4
R7	L2	R31	L4
R9	L2	R3	L1
R52	L2	R74	L1
R53	L2	R75	L1
R55	L2	R76	L1
R60	L2	R77	L1
R63	L2	R78	L1
R64	L2	R82	L1
R65	L2	R83	L1
R66	L2	R84	L1

R67	L2	R85	L1
R70	L2	R86	L1
R71	L2	R87	L1
R72	L2	R88	L1
R73	L2	R89	L1
R79	L2	R91	L1
R81	L2	R30	L1
R58	L2	R48	L1
R35	L2	R56	L1
R39	L2	R97	
R40	L2	R98	

Table 5 shows different scenarios consider while assigning prioritized FRs of Table 3 to two priority groups of different sizes. From Table 5 we can see that total dependencies are reduced from 80 to 41 only when prioritized requirements are assigned to two priority groups of 7 and 93 requirements each. These 7 requirements are top priority requirements which reduce number of dependencies for other requirements of low priority group of 93 requirements. We can see that when prioritized FRs are assigned to two priority groups with 20 and 80 requirements each, total dependencies reduced to 29 only. Dependency reduction column shows number of requirements reduced with prioritization using priority groups for different scenarios considered.

Table 5: Different scenarios consider while assign FRs to Priority Groups

Scenario	Requirements in first group	Requirements in second group	Total dependencies	Dependency reduction (%)
Scenario 1	7	93	45	35
Scenario 2	10	90	37	43
Scenario 3	15	85	28	52
Scenario 4	20	80	25	55
Scenario 5	25	75	31	49

### 6. CONCLUSION AND FUTURE WORK

In this research study, FRs of ODOO ERP as case are efficiently prioritized using CV and Priority Groups with less time complexity. For developers, prioritization of FRs is necessary as one requirement can only be implemented when pre-requisite FRs are implemented first. Also, as in case of parallel development, prioritization of FRs is necessary in order to reduce dependencies among FRs of parallel developers so that timely implementation of requirements can be assured. In this research, 100 FRs of ODOO belong to different modules are prioritized. Results are analyzed for different scenarios of two priority groups which shows significant reduction of dependencies in FRs inside each group. In future, this FRs prioritization approach using CV and priority groups will be evaluated using parallel development scheduling models such as RCPSP by calculating time estimation of each requirements using effort estimation model and whole project. Significant difference in total time estimation of software project with prioritized and un-prioritized FRs will shows advantage of prioritization with technique such as CV with less time complexity.

#### References

- [1] M. Aasem, M. Ramzan, A. Jaffar, and E. S. Islamabad, ‘Analysis and optimization of software requirements prioritization techniques’, 2010.
- [2] M. Yaseen, A. Mustapha, and N. Ibrahim, ‘An Approach for Managing Large-Sized Software Requirements During Prioritization’, 2018 IEEE Conf. Open Syst., pp. 98–103, 2019.
- [3] N. Misaghian and H. Motameni, ‘An approach for requirements prioritization based on tensor decomposition’, *Requir. Eng.*, 2016.
- [4] M. Yaseen, N. Ibrahim, and A. Mustapha, ‘Requirements Prioritization and using Iteration Model for Successful Implementation of Requirements’, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 121–127, 2019.
- [5] M. Ramzan and M. A. Jaffar, ‘Value Based Fuzzy Requirement Prioritization and its Evaluation Framework’, pp. 1464–1468, 2009.
- [6] N. Garg, M. Sadiq, and P. Agarwal, ‘GOASREP : Goal Oriented Approach for Software Requirements Elicitation and Prioritization Using Analytic Hierarchy Process’, pp. 281–287, 2017.
- [7] M. A. A. Elsood and H. A. Hefny, ‘A Goal-Based Technique for Requirements Prioritization’, 2014.
- [8] F. Dalpiaz, ‘Contextual Requirements Prioritization and Its Application to Smart Homes’, vol. 1, pp. 94–109, 2017.
- [9] M. I. Babar, M. Ghazali, D. N. A. Jawawi, S. M. Shamsuddin, and N. Ibrahim, ‘Knowledge-Based Systems PHandler : An expert system for a scalable software requirements prioritization process’, *KNOWLEDGE-BASED Syst.*, 2015.



- [10] P. Chatzipetrou, L. Angelis, P. Roveg?rd, and C. Wohlin, 'Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework', Proc. - 36th EUROMICRO Conf. Softw. Eng. Adv. Appl. SEAA 2010, pp. 361–370, 2010.
- [11] M. Yaseen, A. Mustapha, and N. Ibrahim, 'Prioritization of Software Functional Requirements : Spanning Tree based Approach', vol. 10, no. 7, pp. 489–497, 2019.
- [12] M. Yaseen, I. Journal, M. Yaseen, A. Mustapha, M. A. Salamat, and N. Ibrahim, 'International Journal of Advanced Trends in Computer Science and Engineering Available Online at <http://www.warse.org/IJATCSE/static/pdf/file/ijatcse09912020.pdf> Prioritization of Software Functional Requirements : A Novel Approach using AHP and Spanning Tree', vol. 9, no. 1, 2020.
- [13] J. Ali Khan, I. Ur Rehman, Y. Hayat Khan, I. Javed Khan, and S. Rashid, 'Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique', Int. J. Mod. Educ. Comput. Sci., vol. 7, no. 11, pp. 53–59, 2015.
- [14] N. Kukreja, S. Swaroop, B. Boehm, and S. Padmanabhuni, 'Value-Based Requirements Prioritization : Usage Experiences', Procedia Comput. Sci., vol. 16, pp. 806–813, 2013.
- [15] S. S. Payyavula and F. Si, 'Application of Value Based Requirement Prioritization in a Banking Product implementation', pp. 157–161, 2012.
- [16] A. Herrmann and M. Daneva, 'Requirements Prioritization Based on Benefit and Cost Prediction : An Agenda for Future Research', no. iv, pp. 125–134, 2008.
- [17] M. Daneva and A. Herrmann, 'Requirements Prioritization Based on Benefit and Cost Prediction : A Method Classification Framework', pp. 240–247, 2008.
- [18] A. Perini, F. Ricca, and A. Susi, 'Tool-supported requirements prioritization : Comparing the AHP and CBRank methods', Inf. Softw. Technol., vol. 51, no. 6, pp. 1021–1032, 2009.
- [19] M. Aasem, M. Ramzan, A. Jaffar, and E. S. Islamabad, 'Analysis and optimization of software requirements prioritization techniques', 2010.
- [20] M. Yaseen, A. Mustapha, and N. Ibrahim, 'An Approach for Managing Large-Sized Software Requirements During Prioritization', 2018 IEEE Conf. Open Syst., pp. 98–103, 2019.
- [21] N. Misaghian and H. Motameni, 'An approach for requirements prioritization based on tensor decomposition', Requir. Eng., 2016.
- [22] M. Yaseen, N. Ibrahim, and A. Mustapha, 'Requirements Prioritization and using Iteration Model for Successful Implementation of Requirements', Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 121–127, 2019.
- [23] M. Ramzan and M. A. Jaffar, 'Value Based Fuzzy Requirement Prioritization and its Evaluation Framework', pp. 1464–1468, 2009.
- [24] N. Garg, M. Sadiq, and P. Agarwal, 'GOASREP : Goal Oriented Approach for Software Requirements Elicitation and Prioritization Using Analytic Hierarchy Process', pp. 281–287, 2017.
- [25] M. A. A. Elsood and H. A. Hefny, 'A Goal-Based Technique for Requirements Prioritization', 2014.
- [26] F. Dalpiaz, 'Contextual Requirements Prioritization and Its Application to Smart Homes', vol. 1, pp. 94–109, 2017.
- [27] M. I. Babar, M. Ghazali, D. N. A. Jawawi, S. M. Shamsuddin, and N. Ibrahim, 'Knowledge-Based Systems PHandler : An expert system for a scalable software requirements prioritization process', KNOWLEDGE-BASED Syst., 2015.
- [28] P. Chatzipetrou, L. Angelis, P. Roveg?rd, and C. Wohlin, 'Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework', Proc. - 36th EUROMICRO Conf. Softw. Eng. Adv. Appl. SEAA 2010, pp. 361–370, 2010.
- [29] M. Yaseen, A. Mustapha, and N. Ibrahim, 'Prioritization of Software Functional Requirements : Spanning Tree based Approach', vol. 10, no. 7, pp. 489–497, 2019.
- [30] M. Yaseen, I. Journal, M. Yaseen, A. Mustapha, M. A. Salamat, and N. Ibrahim, 'International Journal of Advanced Trends in Computer Science and Engineering Available Online at <http://www.warse.org/IJATCSE/static/pdf/file/ijatcse09912020.pdf> Prioritization of Software Functional Requirements : A Novel Approach using AHP and Spanning Tree', vol. 9, no. 1, 2020.
- [31] J. Ali Khan, I. Ur Rehman, Y. Hayat Khan, I. Javed Khan, and S. Rashid, 'Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique', Int. J. Mod. Educ. Comput. Sci., vol. 7, no. 11, pp. 53–59, 2015.
- [32] N. Kukreja, S. Swaroop, B. Boehm, and S. Padmanabhuni, 'Value-Based Requirements Prioritization : Usage Experiences', Procedia Comput. Sci., vol. 16, pp. 806–813, 2013.
- [33] S. S. Payyavula and F. Si, 'Application of Value Based Requirement Prioritization in a Banking Product implementation', pp. 157–161, 2012.
- [34] A. Herrmann and M. Daneva, 'Requirements Prioritization Based on Benefit and Cost Prediction : An Agenda for Future Research', no. iv, pp. 125–134, 2008.
- [35] M. Daneva and A. Herrmann, 'Requirements Prioritization Based on Benefit and Cost Prediction : A Method Classification Framework', pp. 240–247, 2008.
- [36] A. Perini, F. Ricca, and A. Susi, 'Tool-supported requirements prioritization : Comparing the AHP and CBRank methods', Inf. Softw. Technol., vol. 51, no. 6, pp. 1021–1032, 2009.