International Journal of Advances in Software Engineering & Research Methodology– IJSERM
Volume 4 : Issue 1 [ISSN : 2374-1619]
Publication Date: 06 April, 2017

A Comparative Analysis on Software Requirements Prioritization Models

[Ataur Rahman, Arif Raza, Muhammad Babar and Fahim Arif]

***Abstract*— Requirements elicitation is the most important phase during requirements engineering process in which requirements are extracted from the stakeholders. One of the vital requirements elicitation activities is requirements prioritization. Requirements prioritization is the process of selecting most significant requirements out of identified requirements. Many requirements prioritization schemes are available in literature, but none of them is considered as a standard technique. The main cause is the parameters that each technique uses. The purpose of this study is to summarize the existing techniques based on the parameters/aspects used in them. Moreover this research study is providing a big picture of prioritization models which is helpful for the researchers working in this area**

***Keywords*— Requirements prioritization, prioritization Aspects, prioritization models, AHP.**

I. Introduction

Requirements elicitation is generally considered as a vital step towards delivering successful software [18][19]. It is the process of understanding, finding, extracting, evolving and discovering needs of stakeholders. Fault, error or bug in this phase can result failure of the other phases and as a result the software project fails. Many of the projects have been failed due to problems in the requirements phase [7][20]. In case of enormous set of requirements, it becomes critical to choose which set of requirements to be implemented first [4][5][6].

Assigning importance or values to the requirements is called requirements prioritization. Requirements prioritization is one of the vital activities of requirements elicitation which cannot be skipped. The key objective of requirements prioritization is to choose the most important requirements from the given set of requirements [15]. Requirements analysis and prioritization is bridge to requirements design which consequently leads to-requirements implementation. Requirements prioritization also assists in identifying requirements abstraction, omissions and ambiguities [8]. To develop quality software it is essential to select right requirements from the given identified set of requirements [1]. The purpose of requirements prioritization is to distinguish the most important requirements for a release [5]. As customer is the primary stakeholder of any software thus the objective of any software is to prioritize the customer needs [6].

Right requirements can be selected through requirements prioritization techniques such as AHP (Analytical Hierarchal Process), grouping, ranking and 100 dollar scheme etc [1]. Prioritization is carried out keeping some parameters /aspects (such as cost, importance to user, isk, time etc.) in mind [4][5][10][14]. Literature shows that different prioritization aspects are important for requirements prioritization. However it is not practical to consider all aspects during the prioritization, somewhat it depends on a specific situation [8]. For example eXtreme Programming (XP) considers only aspect i.e. business value as defined by the customer [5]. In a real time situation, it is always essential to merge different dimensions of requirements together before deciding whether to implement it directly, afterwards or simply not to incorporate it in the system at all [8].

This study discusses the available requirements prioritization techniques and their applicability in different context.

This research paper is organized as: Section 1 is the brief introduction, section 2 discusses existing prioritization models section 3 presents analysis, section 4 discusses future work and conclusion.

II. Prioritization Models

There is large set of prioritization models available, one or few of which are selected for prioritizing requirements. These models are selected based on the nature of the project and its appropriateness for the project.

There are four scales for prioritizing requirements: nominal scale, ordinal scale, interval scale and ratio scale [10]. Two of them are very common i.e. ordinal and ratio scales. In ordinal scale, requirements are ordered to see that which requirements are more critical [1]. In ratio scale decision is taken on comparing a requirement to all other requirements [1]. Ordinal scale, is less influential than that of ratio scale because in ratio scale, each requirement is compared to each other requirement. In existing prioritization schemes some have priorities associated with each requirement, while others group them in priority level [8].

AHP (analytical hierarchal process), cumulative voting, grouping or numerical assignment, ranking, top-ten

Ataur Rahman *(Author)*
National University of Science and Technology
Islamabad Pakistan

Fahim Arif
National University of Science and Technology
Islamabad Pakistan

Arif Raza
National University of Science and Technology
Islamabad Pakistan

Muhammad Babar
National University of Science and Technology
Islamabad Pakistan

requirements, binary search tree, planning game, hierarchal cumulative voting (HCV), value oriented prioritization (VOP), criteria based requirements prioritization extension of HCV, and an extension of AHP for market driven products are some of the prioritization schemes available in the existing studies.

A. Analytical Hierarchal Process (AHP)

Analytical hierarchical scheme is an organized scheme which is used for assigning right priorities and making right decision [15][17]. This scheme is based on ratio scale. Pair wise comparison is carried out between any two requirements in order to recognize the one at high priority [1] [20]. Steps of the technique are as:

- List all requirements in n x n matrix in such a way that it represent all the unique pairs.
- Compare all the unique pairs using ratio scale.
- Relative weight and priority vector are used for these requirements.
- Compute how consistent the prioritizing person has been in his/hers decision.
- The consistency matrix shows the reliability of the result and the judgmental errors that has been made by the prioritizing person.
- If any judgmental error is identified go to step-1, otherwise prioritize requirements [17, 25].

If there are 5 requirements, prioritization matrix will be given as Table 1

Where R1, R2, R3, R4 and R5 are requirements and P1, P2, P3, P4 and P5 are the comparative priorities. There are some confines of this approach such as: it is not efficient when there is large set of requirements to prioritize. For example we have more than 50 requirements, making 60 x 60 matrix and cross comparing become hard. O(n2),quadratic growth occurs in requirements comparison, when the numbers of requirements are increases. It is more difficult to use because n(n-1)/2 comparison needs to be carried out at each hierarchy level [15][17][25].

B. Numerical Assignment

This technique is also called grouping. It is a very famous and essay prioritization scheme which is suggested in [8]. This technique is based on ordinal scale. Each requirement is given a value representing the requirement's supposed significance [7]. There are many variants of this technique exist in literature. Three groups are made i.e. critical, standard and optional. Other names (shown in Table 2) used for the same groups in literature are: high, medium and low [8], or mandatory, desirable and inessential [2].

Table 1: Requirements Prioritization

Requirements	R1	R2	R3	R4	R5
R1	P1	P2	P3	P4	P5
R2	P2	P1	P3	P4	P5
R3	P1	P2	P3	P4	P5

Table 2: Requirements Groups [4][21][22]

S.No	Name	Meanings
1	High/ must have/ Essential/ mandatory/ Critical	Needed for the next release
2	Medium/ should have/ Conditional/desirable/Standard	Assist necessary operation of the system
3	Low/ would be nice to have/ Optional/ inessential/ Optional	Improve the quality, will be nice to have

The number of groups can vary depending upon the situation [1]. For large set of requirements it is necessary to classify them into different groups consistently. Groups provide proper abstraction for prioritization [16]. The abstraction reduces time, complexity and cost of prioritization [15][26]. This technique also provides litheness of increasing or decreasing number of requirements groups [10].

The major problem of this technique is dividing requirements into three groups [1]. In this case each stakeholder will put his requirements in critical group by considering that everything is important. Requirements' having same priority level/group is another problem, because in case of same priority group, reprioritization needs to be done [1].

C. Cumulative Voting

This scheme is also called 100 dollar test or generally voting scheme in literature [1][10][15]. 100 dollar test method was proposed by Leffingwell and Widrig [10]. It is a simple technique in which stakeholders are given 100 unreal votes to distribute among the requirements being prioritized [5]. Each participant assigns votes to requirements according to its criticality and importance. Afterwards, votes of each requirement are summed up, and looked for the requirements that have got highest number of votes [9]. Cumulative voting is based on the ratio scale [20]. The steps involve in this particular technique are: put all requirements in a row; assign 100 votes to all members. Votes are distributed based on significance of a requirement to the product [9].

There are some limitations of this technique as well. This technique fails when a member gives all of his/her votes to a requirement which is important to them but might be inessential for others [8]. It also fails when there are too many requirements to prioritize (more than 100) [1]. This is overcome in literature by enlarging the number of votes from 100 to 10000 [8]. Another limitation of this technique is that the person performing prioritization might miscalculate and the sum may come greater or less than 100 [8]. Furthermore this technique could only once be applied in each project [9].

D. Ranking

In this, requirements prioritization scheme the first most important requirements is ranked 1, second most important is ranked 2 and so on. The least important requirement is ranked as n. It is based on ordinal scale i.e. it is not clear to see the relative difference between any two

requirements like AHP [4, 15]. In this technique each requirement has assigned a unique rank unlike grouping. For obtaining ranks of requirements sorting algorithms such as bubble sort, binary search tree and quick sort are used [8]. This scheme is appropriate when only one stakeholder prioritize requirements. It is suitable to only small projects [4, 15]. Steps of this technique are: first all requirements are listed by a stakeholder than based on the importance, value 1 is given to most critical requirement and n is given to least significant requirement.

E. Top-Ten requirements

It is one of the simplest techniques in term of sophistication and very coarse in term of granularity [4][15][18]. In this requirements prioritization scheme top-ten requirements are picked from large set of requirements without assigning internal order between requirements. This technique does not follow any scale [1]. This scheme is useful in case when multiple stakeholders have equal value [8]. One of the disadvantages of this technique is that no internal order is assigned between the top-ten requirements [4][15]. Consequently decision making during implementation becomes a challenging task. Another challenge to this scheme is to balance issues related to the fact that top priority requirements of all stakeholders are included in the next development activity [1]. Internal prioritization needs to be done among the top-ten requirements in this scheme, because conflict will arise if one or two stakeholders give 1st priority to one requirement and other 2 or 3 stakeholders give 5th priority to that requirement [1]. This issue can also be resolved by trading off between different stakeholders.

F. Binary Search Tree (BST)

In binary tree each node has maximum of two nodes [11][12]. Binary search tree is a unique type of binary search where nodes are tagged with objects [11]. Under the requirements prioritization domain, each object will be a requirement. All the requirements sorted in left sub tree will be less critical/ essential than root node and will have low priorities, while requirements sorted on right sub tree will be more critical/ essential than the root node and will have high priorities [11][12]. Child node also as parent or root node if it has further children. Child nodes having no further children are called leaf nodes [9]. If the requirements are traversed in InOrder then the requirements will be listed in sorted order. If there are n requirements to prioritize then number of nodes in binary search tree will also be n [11].
Prioritizing with binary search tree scheme involves three steps: list all the candidate requirements, select a single requirement as a root node based on which binary search tree is made, and traverse the requirements in InOrder and list them for implementation [11]. In second step child requirements are compared with root requirement if it is more critical than root requirement it will go to the right sub tree other wise to the left sub tree. This step is repeated for each parent node. This approach limits the number of comparisons of AHP by dividing requirements into two sub trees from the start.

G. Planning Game (PG)

PG prioritizes requirements by negotiating with the users /customers. Agile methodologies such as Scrum, XP etc are based on streamline approaches, and try to remove the overhead by reducing documentation. XP is composed of twelve basic practices, one of which is PG [6].

PG is used in XP projects to plan and decide what to develop [6]. In this technique requirements are written on story cards [6][12]. In PG, the customer divides requirements into three different groups. The groups should have the names: those without which the system will not function, those that are less essential but provide considerable business value and those that would be good to have [9]. Meanwhile, the developers estimate the time needed to implement each requirement and, moreover, sort the requirements by uncertainty into three piles: those that they can estimate precisely, those that can estimate reasonably well, and those that they cannot estimate at all [6]. Based on these estimates, the requirements in each pile are prioritized to be implemented in next release [6].

This technique is based on ordinal scale [9]. It is a variation/ combination of numerical assignment technique and ranking [1][12]. Karlsson et al [3] identifies that PG is easy to use scheme. They conclude that both methods have their own advantages and disadvantages, and it would be motivating to have PG to first sort out the most important requirements, and then using AHP to prioritize those requirements [12].

H. Hierarchal Cumulative Voting (HCV)

HCV was developed by Barender and Jonnson [14] to overcome the limitations of AHP and cumulative voting and to integrate the advantages of both the techniques into a single one. HCV is based on ratio scale [13][20]. Ratio scale provides the opportunity to compute the worth of a set of requirements by collectively adding their priorities [13]. By prioritizing through HCV, multiple aspects are taken into account just like AHP [13].

One of the known uses of HCV is that it allows making difficult decisions. In HCV requirements are divided into low and high level requirements. This makes prioritization easy and simple [14]. Risk of neglecting any requirements is also reduced [13]. HCV is composed of the following steps [14]: By using cumulative voting, priorities are assigned to all requirements on related level hierarchy, compute intermediate priorities for requirements, compute final priorities for all requirements relative other and if more than one stakeholder has prioritized requirements, then individual results are weighed together. This step is optional. The major limitation of this technique is that it only considers the “value” aspect during prioritization and ignores other business perspectives, which can lead to wrong results [13].

I. Value Oriented Prioritization (VOP)

It is a good scheme for prioritizing requirements. This scheme is also based on ordinal scale [15]. The key concept of VOP is that it evaluates requirements based on its impact on core business value [13][15]. It also handles the identification and weighting of business uncertainties and implementation cost of every requirement [13]. Using the core business values, requirements values and uncertainties it constructs a prioritization matrix [15]. The matrix contains priorities for all requirements. Requirements having high

priorities are candidates for implementation in the current version/ release [15]. A requirement attractiveness / priority is increased by increasing the worth it provides and decreases by increasing cost, related risk, impact and market factors [13]. There are some aspects which are not considered in this scheme for example the effort and resources necessary for each requirement [15]. This can be counted as its limitation.

J. Model Driven Requirements Prioritization Model (MDRPM)

The MDRPM uses the concept of bins, which is nothing but an ampule to clamp closely related requirements. It uses ratio scale for prioritization. Instead of putting all the requirements to AHP as input, MDRPM make groups of requirements and put to them to modified AHP engine as input which increase the scalability and reduce time as compared to AHP [9].

III. COMPARATIVE ANALYSIS AND DISCUSSION

Based on the above discussed techniques and the studied literature the models have been compared in Table 3 in the appendix, keeping scale, prioritization groups, consistency matrix, constituency of AHP, major advantage and disadvantage criteria in mind.

Requirement Prioritization is the most vital phase in requirement engineering practice. If a project is deprived of assigning accurate and proper requirement to each release, it is almost difficult to complete the project on time and within budget. Assigning discreet priorities is significant for completing a project successfully.

Schemes for establishing priorities have a great significance in software development process. To develop successful and quality software an approach is needed to cover the correct prioritization of the requirements. Therefore this research study has discussed and characterized ten different schemes to establish priorities. The above detail shows that AHP has a promising role in requirements prioritization. It yields the most trustworthy results which are based on a ratio scale, it is fault-tolerant and it includes a consistency check. But it fails when there is a large set of requirements to prioritize. Grouping scheme is the oldest scheme that provides abstraction but it does not assign any tangible priority value to a requirement. Cumulative voting is good when the project requirements are known because it could be applied only once in the project. It could be applied to software having large set of requirements even 100 to 1000. Ranking could be applied to the software projects having a small set of requirements and in this scheme on a stakeholder prioritize the requirements as it is based on a ratio scale so it also does not assign discreet value to the requirements. None of these top-ten prioritization technique uses any scale which could be its major con. Secondly it does not have any procedure for internal prioritization between the top ten requirements. BST is the best technique because it limits the comparisons of AHP but extra time for sorting the requirements is required. PG is the combination of AHP and Ranking and it is easy to use technique but it adopts the disadvantages of AHP and Ranking. HCV is good because it is an improved version of AHP, but it only considers 'value' aspect during the prioritization. VOP is good because it considers multiple aspects during prioritization but it uses ordinal scale in which no cross comparison across the requirements is carried out. MDRPM is best in term of considering ratio scale and automation but yet it cannot handle requirements coming during development and also adopt disadvantages of AHP.

Thus, alternate approach is required to tackle these problems. This research study proposes a model (Integrated Requirements Prioritization Model) that will integrate existing models for mitigating the flaws and improve the performance of the existing schemes [17]. In this research study AHP and Grouping models have been integrated for improving the overall performance and mitigating the individual limitations.

The working sequence of IRPM is as follows:

A set of 48 (as AHP can process up to 50 requirements at a time) requirements are fetched to the grouping model where they are categorized into initial, second, third and final groups (final means high priority and initial means low priority). The final group requirements are then pass to AHP for further prioritization i.e. which requirement out of the final category requirements, should be implemented first. Requirements in the 3rd group are fetched to final, from second to 3rd and initial to 2nd of grouping model. In AHP engine tangible priorities are assigned to each requirement and are sent for implementation according to the priorities.

IV. Conclusion

Prioritization is one of the vital activities of requirements elicitation. This study provides a big picture of all the famous prioritization techniques. Furthermore in this research study, prioritization aspects of different models along with their strengths, weaknesses and structure have been presented. From these weaknesses of the models we come up with a new prioritization technique which is the integration of grouping and AHP models. Complete details of the model in [17] which integrate the advantages of grouping and AHP and tackle the disadvantages of the models. The model is proposed for open source software but could be extended for other closed source software. The model can be automated by implementing it as web or desktop application in any programing language.

References

[1] P. Berander, A.Andrews *"Requirements Prioritization (Chapter 4)*," in 'Engineering and Managing Software Requirements' by Aybuke Aurum & Claes Wohlin (Eds.), Springer 2005.

[2] P. Laurent and J. C. Huang "Lessons Learned from Open Source Projects for Facilitating Online Requirements Processes" REFSQ, LNCS 5512, pp. 240–255, 2009.

[3] J. Karlsson, K. Ryan, "Supporting the Selection of Software Requirements", Procs of the 8th International Workshop on Software Specification and Design (IWSSD '96), 1996

[4] L. Lehtola, M. Kauppinen "Empirical evaluation of two requirements prioritization methods in product development projects". "In: Proceedings of the European Software Process Improvement Conference", Springer-Verlag, Berlin Heidelberg, pp.161-170, 2004.

[5] D. Firesmith, "Requirements prioritization", "Journal of object technology" pp. 35-47, 2004.

[6] L. karlsson, P. Berander, B. Renell and C. Wohlin, "Requirements prioritization: an experiment on exhaustive pair wise comparison Vs Planning Game prioritization", "Proceedings 8th Conference on Empirical Assessment in Software Engineering, Edinburgh, UK", 2004.

[7] A.S. Danesh, R. Ahmad "Study of prioritization techniques using students as subjects" "International Conference on Information Management and Engineering" IEEE pp. 390-394, 2009.

[8] [8] K.A. Khan, ""a systematic literature review of software requirements prioritization"," database for Master Thesis at Blekinge Instittute of Technology (BTH)". pp. 41–43, Oct 23rd, 2006.

[9] M.A. Iqbal, A.M. zaidi and S. Murtaza, " an international conference on, a new requirements prioritization model for market driven products using AHP", "Data storage and date engineering DSDE", pp. 142-149, 2010.

[10] M. Aasem, M. Ramazan, and M.Jaffar, "Analysis and optimization of software requirements prioritization techniques", "International conference on Information and emerging technologies ICIET", pp. 1-6, 2010.

[11] J. Karlsson , C. Wohlin , B. Regnell , "An evaluation of methods for prioritizing software requirements", " Elsevier, New York", 7 Feb. pp. 939-947, 1997.

[12] V.Ahl "An experimental comparison of five prioritization methods", database for Master Thesis at Blekinge Institute of Technology (BTH)" pp. 4-13, 2005.

[13] A. S. I. Mohamed, B. I. A. El-Maddah, and C. A.M. Wahba, "criteria based requirements prioritization for software products", "Proceedings of the 2008 International Conference on Software Engineering Research and Practice SERP", pp. 587-593, 2008.

[14] P. Berande, & P. Jönsson, " Hierarchical Cumulative Voting (HCV) - Prioritization of requirements in hierarchies"."*International Journal of Software Engineering and Knowledge Engineering" 16(6)*, pp. 819-850. 2006.

[15] A.Iqbal, F.M. Khan, and S.A. Khan "A critical analysis of techniques for requirement prioritization and open research issues"," International Journal of Reviews in Computing IJRIC" pp. 8-18, 2009.

[16] K.E. Wiegers "First thing first: Prioritizing requirements", "Software development" 1999.

[17] A. Rahman, A. Raza "Integrated Requirements Prioritization Model for Open Source Software" database for Master Thesis at National University of Science and Technology (NUST)". pp. 33–40, Oct 23rd, 2013.

[18] J.Vijayashree, Dr.Persis Urbana Ivy, J.Jayashree, "Requirements elicitation framework for cloud applications" International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015 ISSN 2091 -2730

[19] T.Arina, S. Norbrat, G.Martin, "How Cloud Providers Elicit ConsumerRequirements: An Exploratory Study of Nineteen Companies" 21st IEEE International Requirements Engineering Conference, Rio de Janeiro, Brazil, 15 July 2013 - 19 July 2013, 105 114

[20] A.A.Muhammad, "Requirements Prioritization: Challenges and Techniques for Quality Software Development" ACSIJ Advances in Computer Science: an International Journal, Vol. 5, Issue 2, No.20 , March 2016

[21] K.E. Wiegers "First thing first: Prioritizing requirements", "*Software development*" 1999.

[22] R.Zornitza ,D. Maya, H.Andrea "A Conceptual Model and Process for Client-driven Agile Requirements Prioritization", "Research Challenges in Information Science (RCIS), Fourth International Conference",2010

APPENDIX

Table 3 Comparative Analysis of Prioritization Schemes

Model Name	Scale	Groups	Consistency Matrix	Constituent of AHP	Major Advantage	Major Disadvantage
AHP	Ratio	No	Yes	Yes	Fault-tolerant and efficient method in case of less number of requirements	Not efficient when there is a large set of requirements to prioritize
Grouping	Ordinal	Yes	No	No	The abstraction reduces time, complexity and cost of prioritization	Every stakeholder places his/her requirements in critical group considering that everything is essential
Cumulative voting	Ratio	No	No	No	Can be applied to large projects by increasing the number of votes from 100 to 1000	This technique could only once be applied in each project
Ranking	Ordinal	Yes up to some Ext.	No	Yes	Very suitable scheme for small projects	Appropriate when only one stakeholder prioritizes requirements
Top-Ten Requirements	No	Yes up to some Ext.	No	No	This scheme is useful in case when multiple stakeholders have equal value	Internal prioritization needs to be done among the top-ten requirements in this scheme
BST	Ordinal	No	No	Yes	Limits the comparison of AHP	Complex process, extra time for sorting is required
PG	Ordinal	No	No	Yes	Easy to use	Adopt the disadvantages of AHP and Ranking
HCV	Ratio	No	Yes	Yes	Integrate the advantages of AHP	Considers the "value" aspect during prioritization
VOP	Ordinal	No	No	No	Consider multiple aspects during prioritization	There are some aspects which are not considered in this scheme for example the effort and resources necessary for each requirement
MDRPM	Ratio	Yes	Yes	Yes	Automated process	Adopt the disadvantages of AHP