IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Software Requirement Prioritization: A Critical Study on Its Techniques

A. Sandanasamy[1*], R. Thamarai Selvi[2]

[1,2] Department of Computer Applications, Bishop Heber College, Trichy – 620017

[*]*Corresponding Author: santhansamy@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 21/Nov/2018, Published: 30/Nov/2018

Abstract-Software has become an indispensible element of day-to-day life. As users of the software are increasing, the need of the software is also getting increased. Developing software that meets the needs of the users is an cumbersome task for a software developer. Since the complexity of the software has increased, the intricacy of the requirement also has increased. As the market needs meet changes, software development team has to meet the changes in the requirements. Hence the developing team faces the challenge to prioritize the requirements in order to produce a risk-free software. This study describes the techniques and methods used in prioritizing the requirements.

Keywords- software requirement prioritizing, software requirement prioritizing techniques, Analytical Hierarchical Process, Numerical method, Fuzzy logic.

I. INTRODUCTION

The essential stages in developing a software are understanding system's requirements, design of system, development of software, system testing.[1]Requirement Engineering is an essential part of Software Development Life Cycle, which includes identification of requirements (termed as an Elicitation/ Gathering), analysis of requirements, their documentation, validation as well as management of requirements. Requirement prioritization falls under the phase of analysis of requirement.[2] Setting up the priorities for requirements is the process of requirement prioritization and the method to achieve requirement prioritization is called Requirements Prioritization Technique (RPT).[3]

In most software projects, there are various stakeholders who wish to find their requirements included in the final deliverables of the product. These stakeholders can be project managers, product managers, developers and testers, users and customers. Since various stakeholders are involved, there can be numerous requirements involving various amounts of resources. sometimes, many of these requirements may demand conflicting features on the software projects. Not all requirements can be fulfilled with available time and resources.[4] Hence the developing team needs efficient methods for prioritizing requirements .

II. MOTIVATION

A software developing team needs to determine the relative necessity of the requirements. Some requirements are mandatory and some may be more critical than others. When certain requirements are failed to be implemented, there is chance of failure of the entire system. At times, unnecessary requirements have to be eliminated. Thus prioritizing the requirements become necessary. Prioritizing may be defined with following meaning

1. Prioritization by implementation order.
   It is the task of determining the implementation order of the requirements.
2. Prioritization by importance.
   It is determining the order of importance to stakeholder [5]

This paper concentrates on the importance of prioritizing the requirements and various techniques used for prioritizing software requirements. The following are the techniques discussed in this study.

1. Analytic Hierarchy Process (AHP)
2. Numerical Assignment
3. Value Oriented Prioritization
4. Requirements Prioritization using Hierarchical Dependencies
5. Binary Search Tree
6. Fuzzy Logic

III. REQUIREMENT PRIORITIZATION TECHNIQUES

A. *Analytic Hierarchy Process (AHP)*

The analytic hierarchy process (AHP) uses decision-making method. This method compares all unique pairs of requirements to decide which of the two has higher priority and to what extent. It uses a 9 point scale in deciding the importance among the requirements. The scale used for pair wise comparison of R1 and R2 is presented in Table 1.

Table 1 AHP Grading Scale

Grade	Meaning
1	R1 and 2j are equally important
3	R1 is moderately more important than R2
5	R1 is strongly more important than R2
7	R1 is very strongly more important than R2
9	R1 is extremely more important than R2
2,4,6,8	Intermediate values.

If a software project has n requirements, then n·(n-1)/2 pair-wise comparisons are required by a decision maker. When the candidate software requirements are compared, their relative priorities are calculated by inserting n requirements in rows and columns of a matrix with order n. For a pair of requirements R1 and R2, the intensity of importance is inserted in the position of the row of R1 meeting the column of R2. In the diagonal position of the matrix, 1 is inserted since requirements are equally important when compared to itself. [6]

Consider three requirement R1, R2 and R3. Assume that the pair wise comparison of these three requirements have given the following results. R1 is essentially more important than R2(intensity of importance *5).* R3 is moderately more important than R1 (intensity of importance 1/3; reciprocal value). R3 is very strongly more important than R2 (intensity of importance 1/7; reciprocal value). Inserting the relative importance as determined by the pair-wise comparisons in the previous example, the comparison matrix is as follows:

$$\begin{array}{cccc} & R1 & R2 & R3 \\ R1 & 1 & 5 & 1/3 \\ R2 & 1/5 & 1 & 1/7 \\ R3 & 3 & 7 & 1 \end{array}$$

The relative priority is obtained by calculating eigenvalues of the matrix. The resulting priorities are relative and based on the ratio scale. Hence, resulting values always add up to 100%. Using the eigenvalues of the comparison matrix, the resulting priorities of the three requirements in the example are:

Requirement R1 28%
Requirement R2 7%
Requirement R3. 65%

From the example, It is determined the requirement R3 to be most important [6].

B. Numerical Assignment

It is one of the simplest method. It uses the nominal scale where requirements are graded as low, medium and high. An IBM tool called RequisitePro uses must, should, could and won't grades to assess the priorities of requirements.. The number and names of the assessment grades can be decided by the requirement engineer based on the application under consideration. The decision maker has the task of assigning grades appropriately to the requirements. One of the possible problem in this method is many requirements may be assigned to the same grade. The meaning of grades is presented in the below table 2. [7]

Table 2. NA Grades

Grade	Meaning
Low	Desirable but its absence does not affect.
Medium	Fundamental requirement and its absence will cause dissatisfaction
High	Crucial requirement and its absence cause the product unacceptable

C. Value Oriented Prioritization

This method uses business value to prioritize the requirements. It identifies the relationship between business values to prioritize the requirements. Each business value is assigned with a scale. The assigning of scale is based on its importance to the organization.[8] The risk of each requirement is assigned with a negative value. Using the core business values and risks, a prioritization matrix is constructed.

Req.	Business values				Risks		Score
	V1	V2	--	Vn	R1	R2	
R1							
R2		W_{ij}			W'_{ij}		

Rn							

The score is calculated using the following formula

$$Sr = \sum_{i=1}^{n}(Vi\ x\ Wr,i) + \sum_{j=1}^{m} Rj\ x\ W'r,j)$$

Where,
Vi is the weight of business value i
Rj is the weight of risk j.
Wi,j is the weight assigned to requirement ri with respect to business value Vj.
W'i,j is the weight assigned to the requirement ri with respect to Rj. [5]

D. Requirements Prioritization Using Hierarchical Dependencies

Luay Alawneh [8] suggested a prioritization method considering the relationship between the needs of the stakeholder and the derived requirements in the form of use cases and non functional requirements. The main process is depicted in Figure 1.

New requirements are gathered from the stakeholders. Newly elicited Requirements are integrated into the project by computing relativeness and prioritizing. The cost-values method is used to prioritize the requirements based on the values to the project and its cost to implement. The relativeness R between RQi and RQj is calculated as given below.

$$WC(RQ_i)=\sum_n WC(UC_n)+\sum_n WC(NFR_n) \quad (1)$$

$$WC_s(RQ_i,RQ_j)=\sum_k WC(RQ_i)\cap\sum_k WC(RQ_j) \quad (.2)$$

$$\overline{RQ_\imath} = WC\ (RQ_i) - WC_s(RQ_i,RQ_j) \quad (3)$$

$$R_{i,j}(RQ_i,RQ_j) = \begin{cases} \frac{\overline{RQ_\imath}}{\overline{RQ_\jmath}} \\ \frac{1}{\overline{RQ_\jmath}}\ when\ \overline{RQ_\imath} = 0 \\ \overline{RQ}_i\ =\ \ when\ \overline{RQ_\jmath} = 0 \end{cases} \quad (4)$$

Equation 1 calculates the total weight of the requirement $RQ_{i.}$ WC represents the total weight.

Equation 2 calculates the sum of weights WC_s for the shared requirements between RQ_i and $RQ_{j.}$

Equation 3 calculates the weight of the covered requirements leaving out the shared requirements. Equation 4 finds the relativeness R .

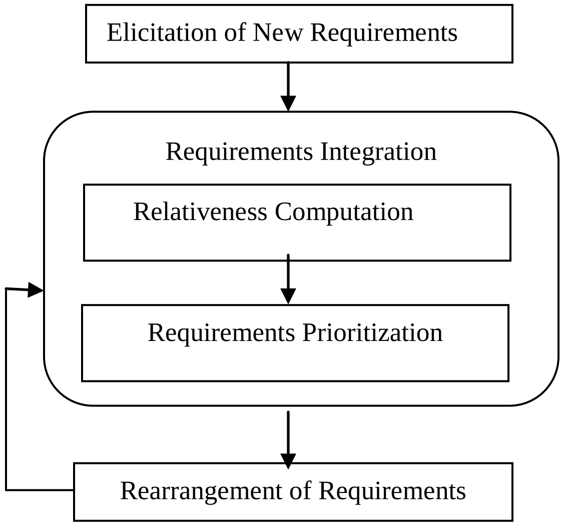
Elicitation of New Requirements
Requirements Integration
Relativeness Computation
Requirements Prioritization
Rearrangement of Requirements

Figure 1. Hierarchical Dependencies Prioritization Steps

The same step is applied to calculate relative value R_v and relative cost $R_{c.}$ This is used as the input for the prioritization step which uses AHP method. [8]

E. Binary Search Tree

A binary tree has two children for every node. Binary search tree falls under type of binary tree. The main property of the binary tree is the right node has greater value than the parent and the left node has smaller value than the parent. Using the same property, the requirement with less priority is placed on the left of the parent and requirement with higher priority is placed on the right of the parent. The steps in arranging the requirements are the following.[9]

1. Take the first requirement and place it at the root node.
2. Take the next requirement and compare with the root node. If it is of lower importance then the root compare with its left child. If it is of higher importance compare with its right child.
3. If the root node does not have any child, the place the new requirement as the child node. If the requirement has low importance than the root, then place it as the left child. If the requirement has higher importance than the root, then place it as the right child.
4. The same process is repeated until all the requirements are placed in the tree. [10]

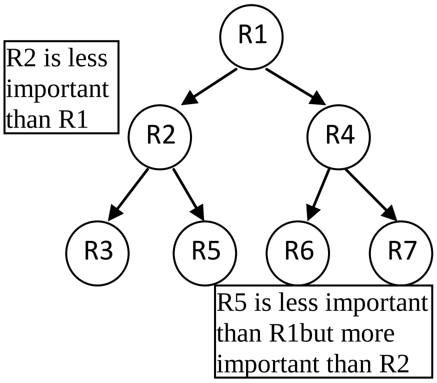
R2 is less important than R1
R1
R2
R4
R3
R5
R6
R7
R5 is less important than R1but more important than R2

Figure 2 Binary Search Tree

When the tree is traversed in in-order, the requirements are listed according to the priority. [10]

F. Fuzzy Logic

Ruby and Balakishan[11] developed a fuzzy logic based prioritization method utilizing customer importance, cost, value and risk. The customer decides on the requirement basing the benefits from the requirement. The developing team decides on the requirement analyzing cost, time and risk from the requirement. Hence both the customers and developers are involved in the prioritizing process.

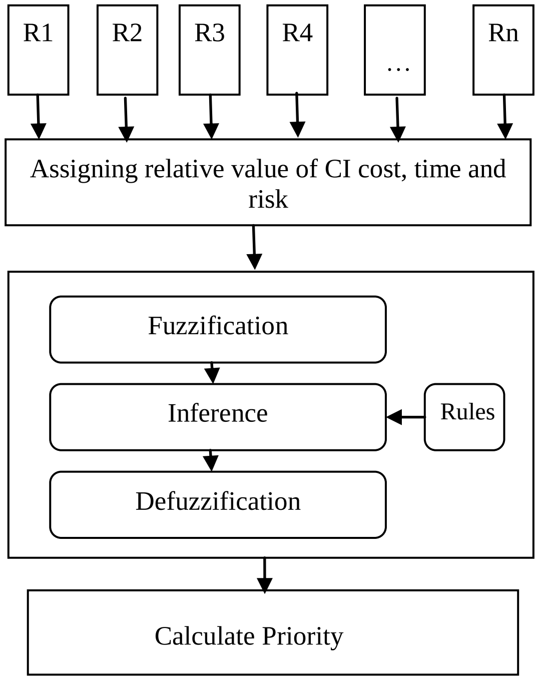

Figure 3. Fuzzy Logic Prioritization

The following are the procedure to prioritize the requirements.
Step 1: The requirements to be prioritized are collected.

Step 2: The value of customer importance is calculated on a scale ranging from 0.0 to 100.0. Value 0 is assigned for lower importance and value 100 is assigned fro higher importance

Step 3: The relative cost to implement each requirement is calculated on a scale ranging from 0.0 to 10.0. Value 0 is assigned for lower cost and value 10 is assigned for higher cost.

Step 4: The relative time to implement each requirement is calculated on a scale ranging from 0.0 to 10.0. Value 0 is assigned for low time and value 10 is assigned for high time.

Step 5: The relative risk associated with each requirement is calculated and the value from 0.0 to 10.0 is assigned.

Step 6: The fuzzification and defuzzification on customer importance, cost, time and risk as input parameter are applied.

Step 7: The priority order is calculated.[11]

IV. SUMMARY

A study has been made in prioritizing the requirements and six techniques have been discussed.. AHP is the oldest method and the slowest among other methods. NA is the simplest method but the accuracy is low comparing the other methods. Fuzzy Logic approach prioritizes the requirements considering the need of both customers and developers. One of the advantage in the fuzzy logic is that once the rules are generated , it could be applied for any size of projects[10]. In Hierarchical Dependencies approach number of comparisons are reduced and it is more accurate than the AHP method[7]. Value oriented prioritization (VOP) technique is considered to be more accurate in generating result while comparing to AHP and NA. Binary Search Tree method requires n log n comparisons and it produces moderately accurate result

REFERENCES

[1] Chetna Sisodiya, Pradeep Sharma, “Scrutiny to Effectiveness of Various Software Process Model”, International Journal of Scientific Research in Computer Sciences and Engineering, Vol.5, Issue.2, pp.88-93, April (2017), ISSN: 2320-7639.

[2] Wiegers, “First things first: Prioritizing requirement,”, Software development Online, vol. 7, pp. 48-53, Sep. 1999.

[3] Karlsson J., Wohlin C., Regnell B., “An evaluation of methods for prioritizing software requirement.” Elsevier, New York, 7 Feb. 1997, pp 939-947.

[4] M. Asaem, M. Ramzan and A. Jaffar, "Analysis and optimization of software requirements prioritization techniques," International Conference on Information and Emerging Technologies, Pakistan, June 2010, pp. 1-6.

[5] J. Azar, R. K. Smith and D. Cordes, “Value Oriented Requirements Prioritization in a Small Development Organization**,**” IEEE Software , 2007, pp. 32-73.

[6] Saaty, T. L., “The Analytic Hierarchy Process”. Mc-Graw-Hill, 1980. .

[7] Persis Voola, A Vinaya Babu, “Comparison of Requirements Prioritization Techniques Employing Different Scales of Measurement**”,** ACM SIGSOFT Software Engineering Notes, July 2013 Volume 38 Number 4.

[8] Luay Alawneh, “Requirements Prioritization Using Hierarchical Dependencies”, Advances in Intelligent Systems and Computing , 2017, pp. 459-464.

[9] Manju Khari and Nikunj Kumar “Comparisons of Techniques of Requirement prioritization,” Journal of Global Research in Computer Science 2013,Volume 4, No. 1, pp.38-43

[10] J.karlsson, C.Wolin and B. Regnell, “An evalution of methods for prioritizing software requirements”, Information and Software Technology, 1997, pp 939-947.

[11] Ruby, Dr. Balkishan, “Fuzzy Logic based Requirement Prioritization(FLRP) - An Approach”, International Journal of Computer Science And Technology, Vol. 6, Iss ue 3, July - Sept 2015**.**