HYBRID APPROACH FOR REQUIREMENT PRIORITIZATION

Haad Ali
Department of Software Engineering
Bahria University, Islamabad

Abbas Ahmad
Department of Software Engineering
Bahria University, Islamabad

Dr. Sheeraz Ahmed
Department of Computer Science
Iqra National University, Peshawar

***Abstract*—The quality of a product is established by the ability to satisfy the needs of the customers and users. Since elicitation of the right requirements and releasing the most appropriate and suitable set of requirements will lead to success of product. If the wrong chunk of requirement is implemented, the user will resist using the software. Requirements Prioritization plays vital role in selecting the 'right' requirements. It is a crucial step, that lead to resolve problems regarding choosing the best set of requirements. The main purpose of the requirement prioritization is to select the 'right' requirements from bunches of requirements. It assists people to select the most desirable requirements. It is considered to be one of the most challenging and difficult task as Lubars stated in his report that none of the companies knew how to assign and communicate priorities. Different stakeholders are involved in assigning priority to each requirement. In requirement prioritization, various aspects are considered such as cost, time, risk, importance, urgency etc. It is very difficult to consider all the aspects for assigning priority. There are many techniques in market for prioritizing requirements such as MoSCoW, voting, numerical assignment etc. but there is not any significant evidence of requirement prioritization technique that will solve the problem of requirement in large chunks. However MoSCow works better than many other techniques for assigning priorities. In this paper, a hybrid approach for requirement prioritization is proposed by combining Important and urgency factors (Eisen Hower Matrix) with MoSCoW. The proposed approach will help to resolve the problem of assigning priorities of requirements in larger sets.**

Keywords*— *Requirement Prioritization techniques, Requirements Prioritization, Challenges in Requirement prioritization.

I. INTRODUCTION

Requirements elicitation is one of the key and vital step of requirements engineering process. It involves a number of activities that are very important for the quality of the product such as requirements discovery, requirements classification and organization, requirement prioritization and requirement specification. Within requirements elicitation, requirements prioritization is one of the most difficult and challenging activity. It is a collaborative, complex decision making activity, which involves a number of stakeholders to assign priority to each requirement and choosing the best set of requirement for next release. It helps to identify the most valuable requirements from user and customer's perspective. Priority is assigned to each requirement by considering several aspects. An attribute or property of a software project defines the aspect, which help to prioritize requirements. Importance, penalty, cost, time (urgency) and risk are most important aspects of a project. One aspect may have an impact on another like a requirement whose priority is high may turn in to less important if it is expensive to be incorporated. Hence it is very important to consider aspect. Although it is near to impossible to consider all aspects. It helps to provide support for different stakeholder to select the best set of requirements, to decide on core requirements of the system, to align business benefits with cost of the requirement, to have an agreement on requirement between different stakeholders and to increase customer satisfaction. In market, there are number requirements prioritization approaches being used but there is not any significant evidence of a technique that will prioritize a large number of requirements. Secondly, most of the techniques like numerical assignment gives same priority to a bunch of requirement, which lead to difficulty in releasing the best set of requirement. Therefore In this paper, we have proposed an approach, in which we have combined two important aspects, importance and urgency that lead to an important matrix named Eisenhower matrix with MoSCoW technique. It will help to assign priority based on different categories i.e. do, decide, delegate and delete. This matrix was proposed for day to day tasks but we have used it in software engineering domain for assigning priority to each requirement. After assigning requirements to these categories, we have used MoSCoW technique to further categorize requirements to get the best set of requirements.

II. LITERATURE REVIEW

There are number of different requirement prioritization methods available that can be used to calculate importance level of each requirement in order to assign different numbers to represent their priority. In this paper, we have examined prioritization techniques and evaluate their applicability in requirement engineering with the aim of classifying most

urgent and important requirements. Some prioritization methods have been describe below in this section.

1) Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process was developed by Thomas L.Saaty in 1970s [3]. AHP is a systematic approach for requirements prioritization of software requirement engineering. Its primary use is to offer solutions to decision problems, it helps the decision makers to identify the priorities and make the best decision possible by comparing a set of evaluation criteria and a set of alternatives options to determine which has higher priority. For comparing in AHP we need a scale form one to nine (1-9) to identify which element is more times important than other elements. Total numbers of comparing elements with AHP are represent as $n \times (n-1)/2$ (where n is the number of requirements).

Table 1. Scale for Pairwise Comparison in AHP [8]

Scale	Value	Explanation
1	Equal value	X and y have Equal importance.
3	Slightly more value	X is slightly more important than y.
5	Essential More value	X is more important than y.
7	Strongly more value	X is strongly more important than y.
9	Absolutely more value	X is absolutely important than y.
2,3,6,8	Intermediate value	Value between the adjacent values.

AHP is a good technique with many advantages like reliability it also have some disadvantages like very slow process due to large number of comparing and also not suitable for large number of requirements[4,5,6].

2) Cumulate Voting

Cumulate voting is also known as the Hundred Dollar (100$) Technique. Cumulate voting is one of the fastest and simple requirement prioritization technique. In this technique the participating stakeholders in prioritization are given a 100 different imaginary units to distribute between the requirements [4]. Study show [4,7] that CV is one of the simple and fastest method, however, this method not perform well and prioritization is miscalculated when there are too many requirements.

3) Numerical Assignment Grouping

The numerical assignment is of the most common and fundamental prioritization technique for software requirements. Numerical assignment is based on dividing requirements into different groups based on their priority, which in some cases may extend number of groups but most commonly used are (High, Medium and Low)[5,7,8,10]. With limited group it leads to confusion of stakeholder between requirements. Study show [6,9] most of the stakeholders think everything is of high priority. If stakeholder prioritize themselves, based on High, Medium and Low categories, stakeholders will most likely consider up to 80 percent of their requirements in high category, 15 in medium and 5 present in low category [6,7]. Another issue with numerical assignment is that number of requirements in each group have same priority level, there is no future prioritization available for each group [6,7,8,9,10].

4) Top-Ten Requirements

Top-Ten also know Top-10 Method is one of the easiest technique for requirements prioritization in the term of sophistication. In this techniques stakeholders are asked to select their top ten priority requirements from their list of requirements [7,11]. Issue with this approach occurs when one stakeholder requirements is given high priority, or mapping more requirements of one stakeholder than others [12], which can lead to dissatisfaction of stakeholders.

5) Five Whys

Five whys is another requirements prioritization technique. It helps to prioritize requirements by asking the stakeholder why requirement is mandatory until the importance of the requirement is established. The question is asked repeatedly (five times or less). It will determine whether a requirement is important or it can be deferred or dropped once the requirement priority is established. This technique has a number of limitations like stakeholder do not help the investigator or different people will come up with different causes of the same problem.

6) Bubble Sort

Bubble sort is closely related to AHP technique, both techniques use pairwise comparisons for ranking requirements [13]. For prioritizing with bubble sort, decision maker will have to arrange all requirements in a vector then comparing two requirements with each other using bubble sort algorithm to determine which of the two requirements is more important, if the lower requirement is more important than the upper one, exchange their positions [7,10,13]. In terms of time, bubble sort is consider to be in the middle. Slower than minimal spanning tree but faster than AHP. Bubble sort is consider to be best for small number of requirements [4,7].

7) Binary Search Tree

Binary search tree is a sorting technique. Requirements in this approach are ranked in a hierarchical order. Each node is followed by two more nodes one on the lift having low priority and one on the right having high priority. In binary search tree each node represent requirement [10] Binary search tree method is illustrated in Figure 1. The main idea of this approach is that each requirement is compare with its parent node, requirement having low priority is places at left side of the parent node and

requirement having high priority is places right side to parent node [5,13].

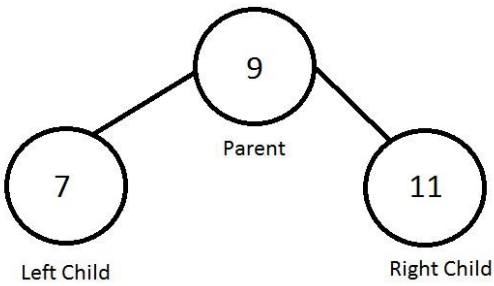


Figure 1. Binary Search Tree

Problem with binary search tree is that it is simply raking requirement in a hierarchical order, but it is not assigning any priority value for each requirement [7,14].

8) Ranking Technique

A simple ranking is based on ordinal scale. This technique implores numbers to rank requirements manually in ascending order form 1 to n (Where n represent number of requirements). The most important requirement is ranked 1st and the least important is ranked n. Problem with ranking technique is that it show lacking when cope with multi stakeholders [6,7] also not it's not providing any details about ranked list or to show any difference between the ranked requirements.

9) MoScoW Technique

MoScoW which is also known as MoScoW analysis is a widely used prioritization technique in requirements engineering. This was developed by Dai Clegg of Oracle UK Consulting in 1994 [15]. It was first used for agile project delivery framework DSDM [16] where focus was on the most important and urgent requirements so that the project is developed in time. For identifying important and urgent requirements which lead to the success of the project, MoScoW is categorized in four priority groups.

a) Must (M):

Must have the urgent and most important requirements which repesents the minimum scope for the product to be useful. These are time critical requirements and faluire of these requirments causes the faluire of the project.

b) Should (S):

Implimentation of these requirements increase product quality but not urgent for the current implimentation of the project. Project's success doesn't rely on these requirements.

c) Could (C):

This group contains requirements that are desireable but not necessary. These requirements are still nice to have if included. According to this method these are the first requirements that have to be removed from the scope, if the project's timescales are at risk.

d) Won't (W):

These requiremnets are least critical to time and success of the project. These are the agreed requirements which will be not provieded in current relaese. Requirements having low importance.

REQUIREMENTS	MUST (M)	SHOULD (S)	COULD (C)	WON'T (W)
Reqwuirent number 1	✓			
Reqwuirent number 2		✓		
Reqwuirent number 3			✓	
Reqwuirent number 4				✓
Reqwuirent number 5	✓			
Reqwuirent number 6		✓		

10) SugarCRM

The SugarCRM technique uses data-mining and machine learning techniques to prioritize requirements based on stakeholders' needs, business goals, system security, and system performance requirements [17]. SurgarCRM utilizes the concept of triage, which is defined as the procedure of determining which of the product requirements will satisfy then given, available workforces, time, budget and other resources [18].

TABEL II REQUIREMENT PRIORITIZATION METHODS

S.NO	Technique	Scale	Limitations
1	Analytical Hierarchy Process (AHP)	Ratio	▪ Not suitable for large requirements. ▪ Very slow process
2	Cumulate Voting, the 100$ test	Ratio	▪ Not suitable for large requirements ▪ Miss calculation ▪ Unequal resource spending.
3	Numerical Assignment Grouping	Nominal	▪ All requirements that are classified into the same group have the same priority. ▪ Categories like high, medium, and low may confuse the stakeholders.

4	MoScoW	Nominal	▪ All requirements that classified in the same class have the same priority ▪ Not suitable for large requirements.
5	Top-Ten Requirements	Nominal	▪ Dissatisfaction of stakeholders.
6	Bubble Sort	Ordinal	▪ Very complex with, projects having medium to large size of requirements.
7	Binary Search Tree	Ordinal	▪ Provides only a simple ranking of requirements without allocating any priority values. ▪ Not suitable for large requirements.
8	Ranking Technique	Ordinal	▪ Shows lacking to multi stakeholders. ▪ Do not show any comparative difference between ranked items.

III. RESEARCH GOAL

The main objective of the proposed approach is to propose a model that will help to prioritize large number of requirements.After a comprehensive literature, it is found that the existing prioritization techniques do not properly prioritize urgent and most important requirements which if not developed on time may delay or fail the project. Therefore, there is a need of a prioritization technique based on the above mentioned factors (Urgent and Important requirements) while prioritizing requirements.

IV. POPOSED MODEL

The proposed approach uses two techniques to prioritize requirements i.e. Urgency and Importance matrix (Eisen hower matrix) and MoSCoW. These two techniques are combined to give better results. It will help to prioritize requirements based on urgency and importance aspect. In first step, the requirements are categorized/plotted by different stakeholders on urgency and importance matrix which consists of four quadrants i.e. do, decide, delegate and delete. The matrix has important and not important on y-axis while urgent and not urgent on x-axis. In “do” quadrant (important and urgent), stakeholder will place requirements that are most desirable, critical requirements that must be included in the system. In “decide quadrant” stakeholder will place requirements that are essential but doesn’t require early delivery. In “delegate” quadrant (urgent but not important), stakeholder will place requirements that will reduce effectiveness if left out while in “delete” quadrant (not important and not urgent), stakeholder will place requirements that are not part of the core but will make the system attractive. The second step is MoSCoW technique of requirement prioritization that will be applied on each quadrant; it’s the most common technique for requirement prioritization. It is based on categorizing requirement in to different group. In MoSCoW technique, “M” stands for “must have” requirements that is not negotiable and must be included in the system, “S” stands for “should have requirement”, if it’s possible we should have it included in the system, “C” stands for “could have requirement”, that is not essential but we could include it if we have extra time and budget and the “W” stands for “would”, that can be left out. It will help to further categorize requirements. This model has the ability to support large number of requirement and each quadrant will have further categorized set of requirement.

S.NO	Quadrant	Requirements Included
1	Do	▪ Essential requirement those must be included in the system. ▪ Dependent requirements. ▪ Critical requirements to achieve functionality. ▪ Deliver early.
2	Decide	▪ Requirements that are essential but does not require early delivery. ▪ Less dependent requirements.
3	Delegate	▪ Requirements are those that would reduce the effectiveness if left out.
4	Delete	▪ Not part of the core, but make system attractive. ▪ Not essential and does not require early delivery.

Steps for prioritizing requirements

1. List all requirements.
2. Estimate their priority.
3. Assign requirements in to each of these quadrants based on their priority and guidelines.

Involved Stakeholders

- Project Manager

- Key Customer Representatives
- Development Representatives

Quad 1 (Important and urgent)

- Essential requirement those must be included in the system.
- Dependent requirements.
- Critical requirements to achieve functionality.
- Deliver early.

Quad 2 (Important but not urgent)

- Requirements that are essential but does not require early delivery.
- Less dependent requirements.

Quad 3 (Urgent but not important)

- Requirements are those that would reduce the effectiveness if left out.

Quad 4 (Not important and not urgent)

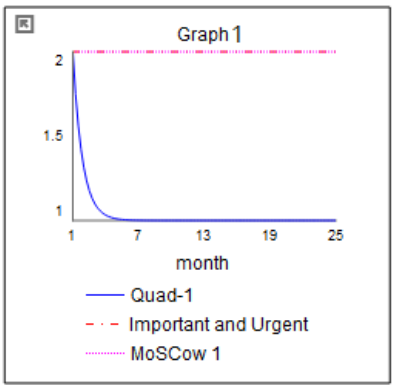
- Not part of the core, but make system attractive.
- Not essential and does not require early delivery.

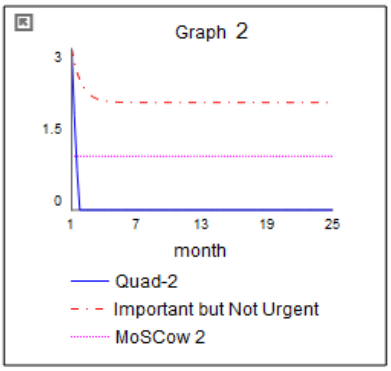
4. After assigning them to each quadrant, use MoScoW technique to further prioritize requirements for each quadrant.

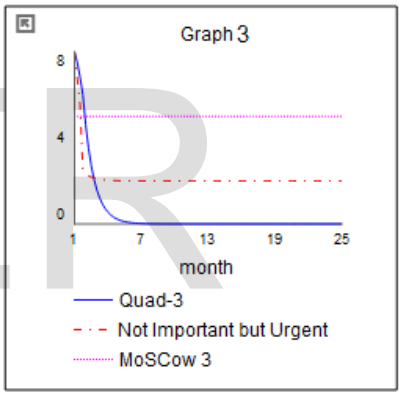
TABLE III PROPOSED MODEL

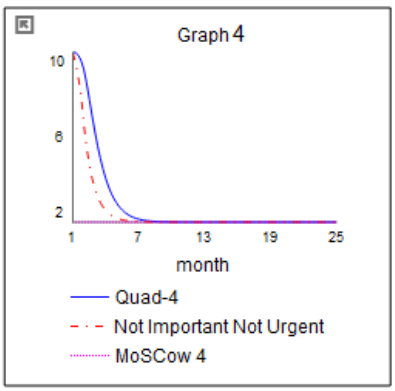
	Urgent	Not urgent
Important	Quad 1 (MoSCoW) Important and urgent	Quad 2 (MoSCoW) Important but not urgent
Not Important	Quad 3 (MoSCoW) Urgent but not important	Quad 4 (MoSCoW) Not important and not urgent

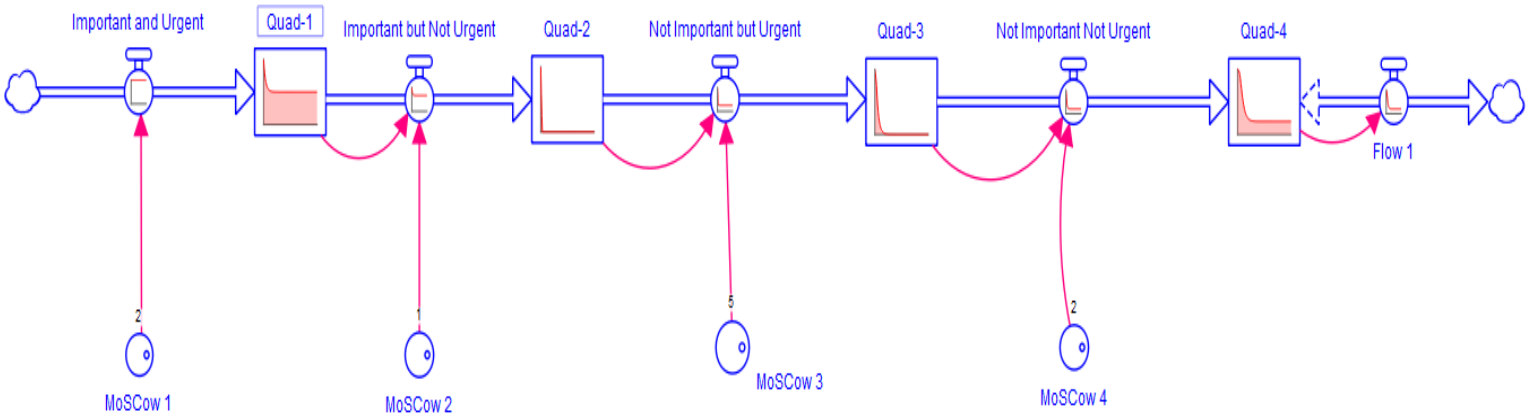
The validation of the proposed model is made via detail simulations using Stella Architect software. The Proposed model is expected to divide requirements in four different quads. The simulation shows requirement prioritization of each quad after filtering out requirements that were contradicting others. (see graph 1) Most important and urgent requirement are grouped into Quad-1. (see graph 2) the results shows those requirements which are important but are not urgent. (see graph 3) this graph represents requirements which are not important but are urgent. (see graph 4) the last graph shows the rejected requirements which are not important as well as not urgent. These changes in requirements and prioritization shows that the quality of dividing requirements of the selected requirements and prioritization increases and is expected to give better results while implementing in agile development.











CONCLUSION

Requirements arises during the entire software development phase and therefore requirements are needed to be managed. As mentioned and discussed in this work, there are many prioritization techniques, proposed frameworks but mostly techniques fail to prioritize large number of requirements, which results in late delivery of software. In the proposed technique, a model has been proposed which will help to prioritize requirements by considering two important aspects i.e. urgent and importance. It merges the effect of two important techniques i.e. urgency/importance matrix and MoScoW, which will aid to categorize requirement more effectively. This approach will also help to categorize large number of requirements. The proposed model has the advantage of prioritizing requirements in large chunks.

REFERENCES

[1] Lubars, M, Potts, C, Richter, C: A review of the state of the practice in requirements modelling.

[2] Bergman B, Klefsjo B (2003) Quality from customer needs to customer satisfaction.

[3] Saaty, The analytic hierarchy process, McGraw-Hill, New York, 1980.

[4] A comparison of nine basic techniques for requirements prioritization M Vestola - Helsinki University of Technology, 2010 - pdfs.semanticscholar.org

[5] Requirements Prioritization Techniques Comparison A Hudaib, R Masadeh, MH Qasem… - Modern Applied …, 2018 - ccsenet.org

[6] Requirements prioritization P Berander, A Andrews - … and managing software requirements, 2005 – Springer

[7] A systematic literature review of software requirements prioritization research P Achimugu, A Selamat, R Ibrahim… - Information and software …, 2014 – Elsevier

[8] Comparison of requirement prioritization techniques to find best prioritization technique JA Khan, IU Rehman, YH Khan… - … Journal of Modern …, 2015 - search.proquest.com

[9] Lausen S (2002) Software Requirements – Styles and Techniques. Pearson Education, Essex

[10] The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study AR Asghar, SN Bhatti, A Tabassum… - International Journal of , 2017 - researchgate.net

[11] Analysis and optimization of software requirements prioritization techniques M Aasem, M Ramzan, A Jaffar - Information and Emerging, 2010 - ieeexplore.ieee.org

[12] A systematic review of software requirements prioritization K Khan - 2006 - diva-portal.org

[13] J. Karlsson, C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements," Information and Software Technology, vol. 39, no. 14-15, pp. 939 – 947, 1998.

[14] A survey of requirement prioritization methods G Kaur, S Bawa - Int. J. Eng. Res. Technol, 2013

[15] Clegg, Dai; Barker, Richard (2004-11-09). Case Method Fast-Track: A RAD Approach. Addison-Wesley. ISBN 978-0-201-62432-8.

[16] Bittner, Kurt; Spence, Ian (2002-08-30). Use Case Modeling. Addison-Wesley Professional. ISBN 978-0-201-70913-1.

[17] Towards automated requirements prioritization and triage C Duan, P Laurent, J Cleland-Huang… - Requirements …, 2009

[18] Davis AM (2003) The art of requirements triage. IEEE Comput 36(3):42–49