International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Software Requirements Prioritization Tool using a Hybrid Technique

Jamilah Din, Ilori Michael, MuhammedBasheerJasser

Abstract: ***Requirements Prioritization (RP) helps to discover the most desired requirements. System developers are not always fully able to implement stakeholders' requirements when time, scope and cost are limited. To solve this challenge, requirements must be prioritized. A few prioritization techniques have been proposed but nonehasbeen automated. Furthermore, rank reversals (updating rank status whenever requirements are added or deleted) is a major limitation of existing techniques. This paper uses AHP and Cost-Value methods for requirement prioritization while addressing rank reversal. The techniques are able to rank every requirement based on relative value and implementation cost. The AHP method allows an input from the system developers and system users and makes a pair wise comparison. The Cost value method ranks the requirements in terms of the ratio of value to cost. As requirements increases, human inconsistency in judgment increases. AHP calculates inconsistency and if it is less than 0.1 then stakeholders should consider new input values. The prioritization methods are implemented into a prioritization tool which creates a list of the prioritized requirements as output, then consequently generates a scatter plot to display relative values and implementation costs of input requirements. This tool is evaluated in terms of efficiency (total time taken for prioritization) and ease of use.***

Keywords: Software Requirements, Requirement Prioritization, Prioritization Techniques.

I.INTRODUCTION

Implementing too many stakeholders' requirements; when time, scope and cost are limited; may cause failure in projects. For that, it is significant to select the right requirements from a set of candidate requirements, because the most cost effective way to develop software is to find the optimal set of requirements early in the development phase. Requirements Prioritization (RP) is distinguishing the foremost essential requirements from a list of requirements as perceived via the stakeholders, with a definitive objective of acquiring a shared rationale for apportioning them into subsequent product releases (Karlsson et.al, 1998).

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Jamilah Din, Faculty of Computer Science and Information Technology/ University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
Ilori Michael, Faculty of Computer Science and Information Technology/ University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
Muhammed BasheerJasser, Faculty of Computer Science and Information Technology/ University Putra Malaysia43400 UPM Serdang, Selangor, Malaysia

The selection of the right requirements is the main challenge to maximize business value, enjoy project success and satisfy the key needs of the stakeholders in terms of implementation costs and relative value.

On the other hand, wrong selection of requirements leads to an increased cost and can have an adverse effect on the quality of the system. Priority is the relative right of a requirement to the utilization of limited resources. If the resources are unlimited, then there will be no need to prioritize requirements, but due to the budgetary deadlines and time to market constraints, it could be a challenge for requirements engineers, system developers to decide which requirements lead to a high stakeholder satisfaction because that should be considered first (Chomal&Saini, 2016). On this premise, RP plays an integral responsibility by addressing high priority requirements before considering the low-priority ones. Therefore, RP is considered highly significant but a complex selection process in making decision throughout the software phases of development. Notably, disagreements, conflicts or breaches in contracts are avoided (Eman et.al, 2016).

The purpose of this work is to aid system developers and system users to prioritize requirements using AHP and Cost value methods. A tool is developed using these methods. As requirements increases, human inconsistency in judgment increases, AHP calculates inconsistency and if it is less than 0.1 then stakeholders should consider new input values. This research contributes to requirements prioritization research community by presenting a Hybrid Prioritization Technique Tool that is implemented for software requirements prioritization.

The rest of the paper is organized as follows: Section 2 discusses the literature and identifies limitations of existing prioritization methods. Section 3 presents the research methodology. Section 4 introduces the proposed architecture. Section 5 discusses the results. Section 6 concludes the research and suggests an area for future work.

II. LITERATURE REVIEW

Some well-known prioritization techniques have been proposed in recent years and they describe how prioritization exercise of requirements can be carried out. The basis of RP techniques lies within scoring strategies which are decision orientated, and a number of these scoring strategies generate relative weights for the requirements that are being considered. Moreover, the relativity of requirements can be calculated out of a pool according to some criteria with certain specific scale. Prioritization techniques can be specifically categorized as nominal scale,

ordinal scale, and ratio scale, as in Figure 1.

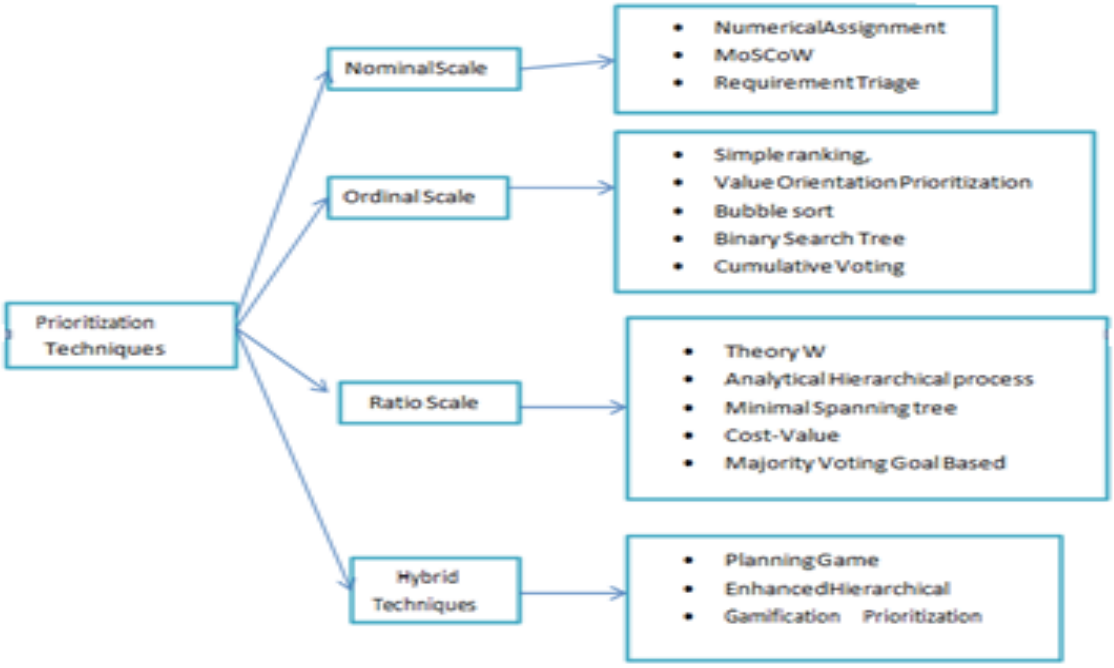

Fig. 1 Existing Prioritization Techniques

The nominal scale assigns each requirement to different priority groups based on perceived importance. This is the lowest dimension of measurement that relies upon grouping requirements without numerical importance.The ordinal scale is quantitative and is used as a comparison parameter to understand whether a requirement is greater or lesser than the other using sorting. The ratio scale compares candidate requirements in a pair-wise manner. The stakeholders thus have to decide which requirement is more important, and to what extent. It is multi-aspect in nature. Hybrid combines two more prioritization techniques; exploit their strengths in other to perform requirements prioritization. For instance, EHRP consists of AHP, QFD, and CV which is a hybrid combination of the three famous RP techniques(Abou-Elseoud et.al, 2016). The challenge of existing techniques is high time consumption, and they are not effective if the project size is large (Babaret.al, 2015). Therefore, considering this situation in system development, then there is a huge need for supporting tools to cater for enormous number of requirements(Mkpojiogu et.al, 2017).

Unfortunately, prioritization techniques exist yet supporting tools that can enable stakeholders to carryout requirements prioritization are not available. Also, there has not been a combination and implementation of AHP and cost value in the literature. The scope of this research is to propose and implement a hybrid technique (AHP and cost-value) that can help system developers and system users to determine preferential requirements for implementation cost and relative value respectively via a weighting scale to get ready for subsequent system releases. Finally, this tool is evaluated in terms of efficiency (total time taken for prioritization) and ease of use as presumed by the stakeholders.

III.RESEARCH METHODOLOGY

This qualitative research methodology shows how the research has been conducted toimplement a tool that can prioritize requirements and rank these requirements considering cost and value.This incorporates an outline of a detailed data gathering techniques and data analysis approach, as demonstrated by the research methodology.The methodology is shown in Figure 2.

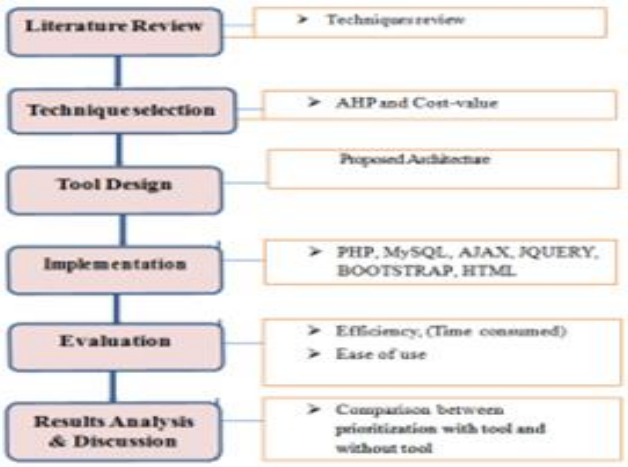

Fig. 2 Research methodology

IV.THE PROPOSED ARCHITECTURE

The architecture of the proposed tool consists of five components/levels,starting from inputting the requirements to outputting the ranks of the requirements, which are: User profiling, Requirements Processing, Database Management, Rank Processingand Output Management. These levels are presentedin Figure 3. In the next subsections, we explain in more details these levels.

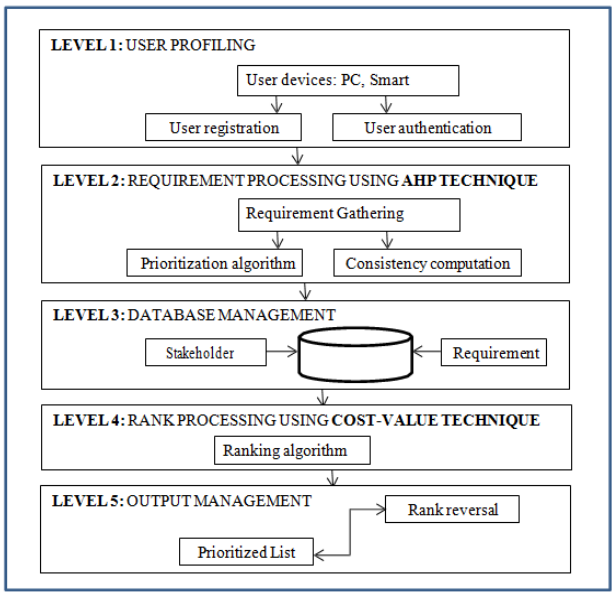

Fig. 3 Architecture of proposed tool

Level 1 (User Profiling)

A stakeholder can communicate with the requirements prioritization tool to perform prioritization exercise from a PC or smart phone or even personal digital assistants through the interface, shown in Figure 4. Stakeholders can only perform prioritization exercise after their devices have connected to the tool through the internet. Stakeholders will have to register their details in other to have their records to know who prioritized what and for privacy reasons as well. Stakeholders can either be developers or system users. Once they register with their password, their passwords are encrypted automatically and stored in the database. Also, users will have to be verified before they can log on into the system to perform the prioritization exercise by using anemail and password.

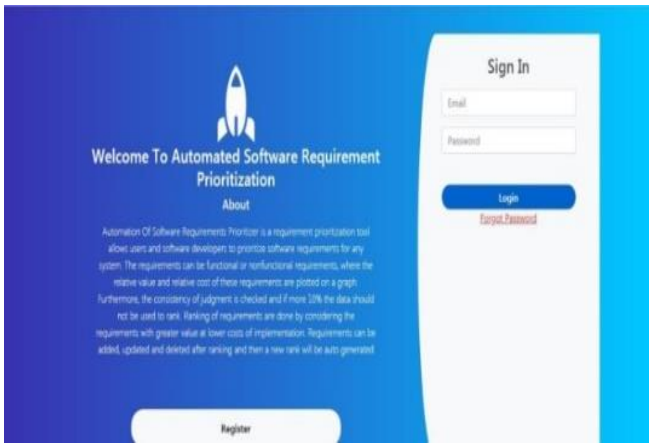

Fig. 4 User profiling interface

Level 2 (Requirement Processing)

Requirement gathering is performed by system developers and they created requirements (functional and non-functional) for a particular project. The prioritization algorithm, for calculating the relative cost and relative value of the requirements given by the stakeholders, consists of the following phases:

Phase 1: Say the prioritization of event cost and value of the added system requirements *R1, R2...R* and stakeholders(developers) *S1, S2...S*, pair wise comparison of these requirements is generated for each stakeholder with added requirements (both functional and non-functional).

Phase 2: By using AHP scale, creates a pair wise comparison for the added requirement to rank every stakeholder S.

Phase 3: Sum up each req. column, for all stakeholders by adding all the values on that column.

Phase 4: Divide each value of each requirement column by sum of that column, and then a generation of the rankings in matrix form is created.

Phase 5: Add values on each row then divide by the amount of requirements. This will derive the relative cost and relative value for developers and system users respectively

It is important to note that the system user can only start the pairwise comparison if the administrator has assigned the project to that user. Once this is completed, the entire functional and non-functional system requirements for that particular project is displayed in pairwise comparisons, as in Figure 5.

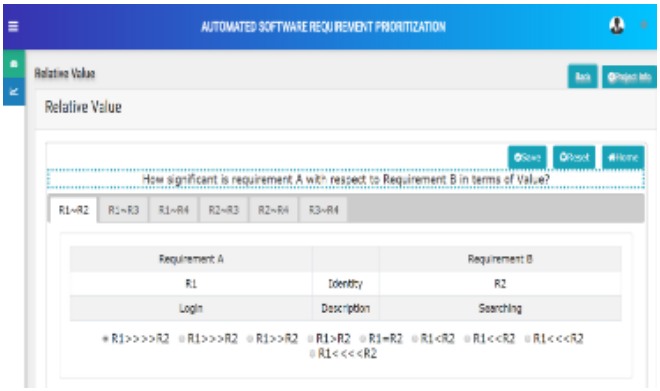

Fig. 5 Requirement comparison weightage

Consistency computation: If Req1 is considered extremely valuable to Req2, Req2 is perceived somewhat more valuable to Req3, and Req3 is slightly more valuable to Req1, then inconsistency has occurred and result's accuracy is reduced logically. This is why calculating consistency ratio of requirements is highly significant and then measure user judgment errors through calculating consistency index of the comparison matrix.Follow these steps to achieve this aim:

Step 1: Comparison matrix should be multiplied by priority vector (relative value and relative cost)

Step 2: Each element in the priority vector should divide each element in resulting vector

Step 3: Calculate the average over the elements in the resulting vector

Step 4: Compute Consistency Index/Relative Index

A rule of thumb indicates that consistency ratio of 0.10 or less is viewed as ideal. But in reality, there is a frequent occurrence of consistency ratios exceeding the acceptable 0.10 benchmark. Figure 6 shows the inconsistency detector interface.

Fig. 6 Inconsistency detector interface

Level 3(Database - management system)

MYSQL server is used to build database management system of the tool. This database organizes the requirements including the configuration needed for the tool to function as expected. Relative value and relative cost computed by the AHP is stored in the database, where the controls of retrieval and update can be done. The database also stores the information of the users i.e. the user profiling level. So basically, both stakeholder's information and requirements computation results are stored in the database. The module database management helps to retrieving data and controls from the database including storing data. Requirements are allotted unique identification numbers individually to avoid duplication.

Level 4(Rank processing)

In this level, the rank is computed, using the relative value and implementation cost that was processed at the requirements stage and stored in the database, as follows:

$$\text{Rank} = \frac{\text{Relative value}}{\text{Relative cost}}$$

Table 1 shows examples of measuring ranks of requirements.

Table. 1 Sample requirements ranking

Requirement	Relative value	Relative cost	Rank
R1	16%	5%	1
R2	10%	7%	3
R3	13%	4%	2

Level 5 (Output management)

The output is displayed after the cost-value has been computed by developers and system users for that project. The developed interface can be integrated with modern information and communication technologies, principally the Internet to intelligently prioritized requirements and display requirements rank. This level has two parts: The prioritized list and the rank reversal.

The prioritized list is the ranking of all the requirements based on the ranking algorithm as described in level 4, from highest to lowest (Descending order). Based on the relative value and implementation cost, the system automatically generates graph displaying the prioritized requirements, as in Figure 7.

The second part of output management is the rank reversal. The pseudocode of the rank reversal is shown in Figure 8. If there should be a need for stakeholders to edit their selections, such as adding or removing requirements, then, RP will self-update requirements ranks then re-compute results again. The outcome of this tool will facilitate system developers and system users in deciding where to make investment of cost and value for every requirement so as to reinforce delivery of fine quality and usable software that meets users' expectations because only then a system is considered successful.

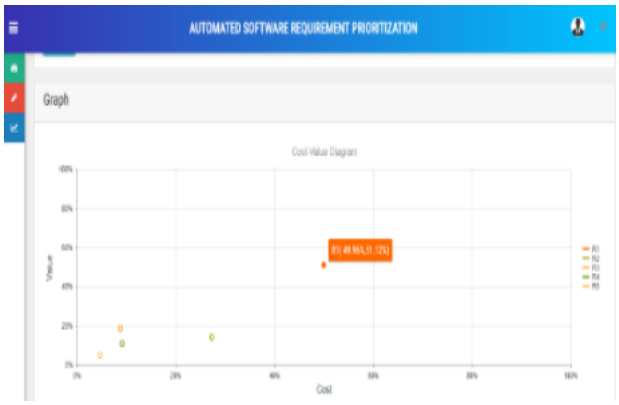

Fig. 7 Scatter plot of value to ratio

```
For each added or deleted requirement,
Add the value to the list of rating.
UpdateQuery Method

Initialize the query statement as details with update statement.
Initialize string variable as rating
For each rating made by the user,
     Add the rating to string variable rating.
Add rating to details.
Add WHERE clause to specify the stakeholder's id and the project id.
```

Fig. 8 Pseudo code 1: Rank reversal approach

V.RESULTS AND DISCUSSION

The proposed system was carried out in the context of ten Masters Students of the faculty of Software engineering to prioritize tenactual requirements consisting five non-functional requirements and five functional requirements of an ATM (Automated Teller Machine). Point of view was from the choice maker's perspective where the subjects were asked to carry out prioritization procedures in other to investigate the difference between prioritization without tool against prioritization using the implemented tool in terms of time consumption and ease of use. Time consumption for prioritization is an objective measure of average time taken by a choice maker to complete prioritization process using a method (Hudaib (2018), CHOMAL(2016)).

This was done by monitoring begin time and finish time of prioritization task using each prioritization method, afterwards compute differential ratio. How easy it takes a choice maker to carry out prioritization process using the proposed tool and without tool measures Ease of use. In this experiment, we measured ease of use by a questionnaire which was handed immediately after working with each prioritization approach.

Total time consumption: It was discovered that the tool-based approach (Automated) was way faster than the Manual-based because it requires 77% less than the manual approach to complete the prioritization process prioritization, as in Figure 9.

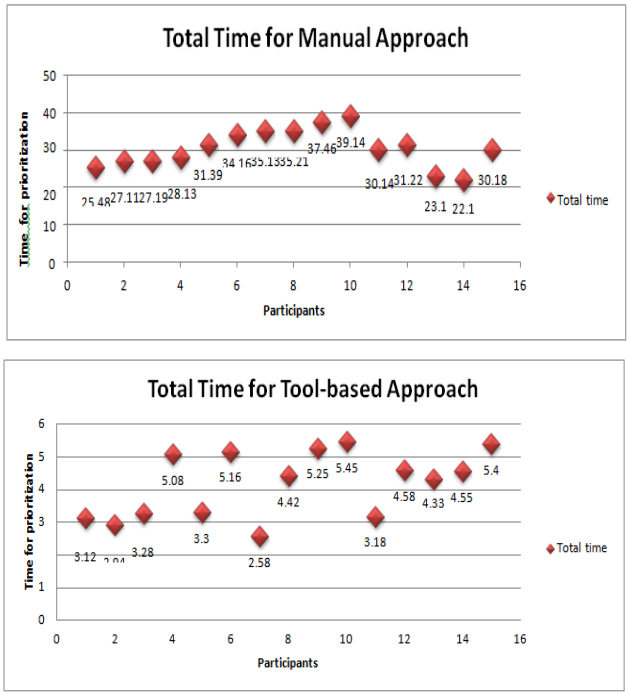

Fig. 9 Total prioritization time

Ease of use: It can be seen from the comparison that 13% of the subjects were neutral about both approaches. The percentage of participants that strongly disagreed to the ease of use of Manual-based is on the high side with 1 participant agreeing that it is easy to use, that is, 33% and 13% respectively. A percentage of 53% of participants strongly agreed that the tool was easy to use as compared to prioritizing without the tool. These results are shown in Figure 10.

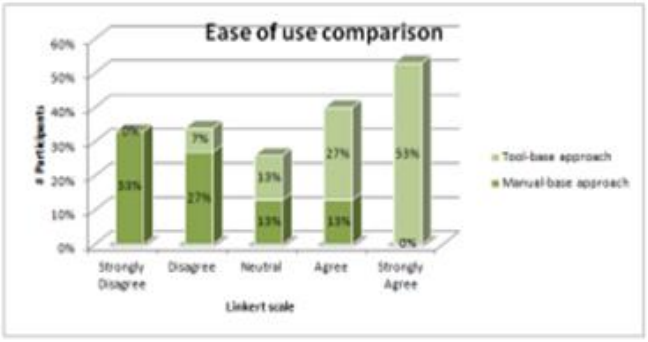

Fig. 10Ease of use for prioritization

VI.CONCLUSION AND FUTURE WORK

By developing a tool for requirement prioritization, it becomes possible to prioritize requirements in terms of implementation cost and relative value at less time while enjoying high ease of use.

In this paper, limitations of existing prioritization techniques have been identified which are scalability, rank reversal, high time consumption and unreliable ranking. However, an improvement has been made on them by proposing a hybrid technique based on Analytical Hierarchical Process and Cost-value to deal with prioritization of requirements in order to generate reliable ranking of requirements in lesser time. From the results discussion, it is clear that prioritization completion success rate of the proposed system approach is more than that of the traditional manual approach.

The implementation is designed to be an online tool so that system developers and system users can directly participate during requirement prioritization. Hencethe tool improves the quality of software product development by reducing software development time while providing an ease of use. The scope of this tool is only limited to requirements that do not depend on each other. We intend to cover this requirements dependency in our future works.

REFERENCES

1. Abou-Elseoud, M.A., Nasr, E.S. and Hefny, H.A., 2016, December. Enhancing requirements prioritization based on a hybrid technique. In 2016 11th International Conference on Computer Engineering & Systems (ICCES) (pp. 248-253). IEEE.
2. Babar, M. I., Ghazali, M., Jawawi, D. N., Shamsuddin, S. M., & Ibrahim, N. (2015). PHandler: an expert system for a scalable software requirements prioritization process. Knowledge-Based Systems, 84, 179-202.
3. Chomal, V. S. & Saini J.R. (2016). A Parameters-based Approach for Requirement Prioritization for Software Development in Academic Context, Conference of ETIT, At Veer Narmad South Gujarat University, Surat, Volume: Vol.5. No. 1.
4. Eman S. Nasr, M., Nasr, E. and Hefny, H. (2016). Enhancing Requirements Prioritization Based on Hybrid Technique. IEEE, 978-1-5090-3267-9, pp.248 - 253.
5. Hudaib, A., Masadeh, R., Qasem, M. and Alzaqebah, A. (2018). Requirements Prioritization Techniques Comparison. Modern Applied Science, 12(2), p.62.
6. Karlsson, J., Wohlin, C. and Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. Journal of Information and Software Technology, 39(14-15), 939-947.
7. Mkpojiogu, E. O., & Hashim, N. L. (2017). Quality Based Prioritization: An Approach for Prioritizing Software Requirements. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2-2), 17-21.