ANALYSIS OF SOFTWARE REQUIREMENTS PRIORITIZATION TECHNIQUES

Manisha Malhotra
Ph.D Scholar, IK Gujral PTU,
Jalandhar City (India)

RPS Bedi
Joint Registrar, IK Gujral PTU,
Jalandhar City (India)

Abstract: During software development, Requirements prioritization is performed to rank the requirements in order of their importance and implementation releases. This paper analyses few requirement prioritization techniques closely and how each one is different from one another on the basis of usage, time, scalability, number of comparisons and accuracy.

Keywords: Requirements prioritization, stakeholders, comparison, software engineering.

I. INTRODUCTION

Software Engineering is a disciplined, systematic and quantifiable approach to the development and maintenance of the software product. Many of the Software development phases involve high communication intensive activity that involves analysts, architects, developers, testers, business stakeholders and end users. Among all the software development phases and products, requirements specification is the key ingredient in the process of designing, visualizing and developing any system. Larger projects are more likely to fail and could have very high maintenance cost, if the requirements are not clearly understood and managed.

Requirements prioritization is one of the critical steps in software development process. Requirements prioritization is introduced first to present the role of requirements prioritization in software engineering and how it relates to other critical steps to produce high quality software systems.

Our goal: In this paper, we will compare different software requirements prioritization techniques. The chosen techniques are Analytic Hierarchy Process (AHP), Cumulative Voting (CV), Numerical Assignment Technique (NAT), Binary Search Tree (BST), Planning Game (PG) and Value Based Intelligent Requirements Analysis (VIRP). To study these techniques, we will systematically apply all the techniques to prioritize a set of ten quality requirements. We then categorized these techniques from a stakeholder’s perspective, taking into account five criteria such as ease of use, total time taken, scalability, accuracy and total number of comparisons required to make decisions.

II. MOTIVATION

Projects are still suffering low success rates. According to a report,

- 32% of all the projects are successful, in the sense that they are delivered on time, on budget and with the required features and functions.
- While 44% are described as challenged, which means, they are delivered late and/or over budget and/or have less than the required features and functions.
- Remaining 24% of them are failures, being terminated before completion or delivered but never used.

Main factors causing challenged or failed projects are:

1. lack of user involvement,
2. lack of resources,
3. unrealistic expectations, and
4. Changing requirements and specifications.

So we say that most of the systems fail due to lack of exact definition of requirements and stakeholders’ involvement in requirements prioritization. As such, the requirements should be well understood, well written and well prioritized so as to get success in software development project. And at the same time software developing companies does not know how to prioritize requirements and which technique is best applicable for their software. So keeping all these things into account, this paper will analyze these requirements prioritization techniques.
III. REQUIREMENTS PRIORITIZATION TECHNIQUES

Requirements Prioritization can be done with various different scales and types. Below, few of the prioritization techniques are presented.

1. Numerical Assignment or Grouping
   The most common prioritization techniques, and is based on grouping the requirements. In this technique, requirements are grouped together in different groups i.e. critical, standard and optional. The stakeholders then are asked to classify the set of requirements into these three groups on the basis of their priority.
   - Technically, this technique is medium but very easy in terms of complexity.
   - Usage of the terms like low, medium and high can create confusion for the stakeholders as they may have different perceptions for these relative terms.
   - No clear definition of each term and group.

2. Analytical Hierarchy Process (AHP)
   Analytic Hierarchy Process is a powerful statistical technique based on relative importance of requirements with respect to one another. It uses pairwise comparison matrix to calculate relative importance and the values range from 1 to 9 where:
   - AHP give better results when it is used in linear development model with single stakeholder and with less constrained projects.
   - It is not suitable for iterative model where the requirements are ambiguous.

3. Cumulative Voting or 100 $ Test
   The cumulative voting is a straightforward technique in which the stakeholders are provided 100 imaginary units, which are based on the ratio scale. Each stakeholder has to divide his units in the requirements as per his/her understanding. These imaginary units could have different aspects like money, cost of implementation, penalty, hours, importance etc. The units are distributed among the requirements and the sum total at the end should be hundred. But, this may possess problem if the number of requirements is greater than hundred.
   - This technique is complex in terms of cleverness and fine in terms of granularity.
   - During prioritization process, sometimes prioritization can be miscalculated and the sum may come greater or less than hundred for large number of requirements.
   - This technique works well in all the environments where involvement of stakeholder is vital.
   - This technique is better when the requirements are ambiguous.

4. Binary Search Tree (BST)
   Binary Search Tree method is another technique used for sorting and was suggested by Hopcroft, Aho and Ullman [12]. In this technique, all the buds have mostly two children. Kaarlsson presented binary search tree for requirement prioritization for the first time[5]. Each bud in binary search tree shows a requirement. The low priority requirement are those requirements which are arranged on the left side of bud and high priority requirements are those requirements which are placed on the right side of the bud in binary search tree. We do requirement prioritization such as initially we take a requirement and accommodate that requirement as a base bud. After that we take another random requirement and analyze it to the base bud. If that requirement is negligibly essential than the base bud we then measure it to the left child bud of the root bud and if the requirement is of larger importance than the base bud then we analyze it to the right side child of root buds. The base bud if do not have any child buds
then place that requirement as a new child of the root bud. If the requirement having larger preferences then root bud place that provision as a child of the root bud and if it’s of less preference then root bud place that requirement to the left side bud as a new child of the root bud. This process is repeated all the requirements are placed in the binary search tree.

- This technique is effective where only one stakeholder is there.
- It is suitable where the number of requirements is not quite large.

5. Planning game (PG)
This technique combines Numerical Assignment and ranking prioritization techniques. In this technique, requirements are prioritized in consultation with the stakeholders. The requirements are first prioritized by stakeholders in three different piles as critical, medium and low. And at the same time developers calculate time to implement those requirements distribute them on the basis of risk i.e. those that can approximate 1) accurately, 2) logically and 3) cannot approximate at all.

- This prioritization technique is most suitable to extreme programming.
- It offers more flexibility than numerical assignment where users are asked to divide the requirements into three groups.

6. Value Based Intelligent Requirement Prioritization (VIRP)
VIRP [8] is basically a multilevel prioritization and classification technique. This technique involves the use of stakeholders’, experts’ and automated fuzzy logic based system at various stages to iteratively prioritize the requirements.

- This iterative prioritization ensures that requirements are evaluated again and again by different actors and a more meaningful and realistic result is achieved.
- This technique is time consuming and also complex.
- It is good for medium to large systems.

IV. ANALYSIS OF COLLECTED DATA

**Design:** The testing begins with the first question of every technique; followed by the second and third and so on. For each question, participants ranked each of the prioritization method and finally mean value was taken.

The participants were asked questions on the basis of following criteria:

- Ease of use: How easy the prioritization technique was to apply.

The answer of the question is shown in fig. 1.

![AVG RANK](image)

**Figure 1.** Comparison on the basis of ease of use.

Fig. 1 clearly indicates that participants thought that Planning Game (PG) followed by VIRP was the easiest method to apply. NAT followed by AHP was most difficult to handle. CV and BST were in the middle of these two groups.

- Total time taken: How long time it took for the participants to perform the prioritization with the techniques under consideration.

The result of the question is shown in fig. 2.

![TOTAL TIME TAKEN](image)

**Figure 2.** Comparison on the basis of total time taken.

From the result in fig. 2, clearly NAT took the longest time to execute, followed by AHP. The fastest technique was VIRP and PG. Between fastest group of techniques and slowest group of techniques was CV.

- Scalability: Arrange the methods according to how the participants believed that the methods would work with many more requirements than the 13 that were in the experiment.

The result is presented in fig. 3.
Figure 3. Comparison on the basis of scalability.

The result in fig. 3 indicates most of the participants thought VIRP, and BST were the prioritization techniques that were more suited as candidates to handle much more requirements. The participants found that AHP followed by NAT would be the worst candidate to scale up for more requirements. In the middle was PG.

d. Accuracy: The participants were asked to arrange the techniques according to their opinion about accuracy of the result produced by each method. The result is shown in fig. 4

Figure 4. Comparison on the basis of accuracy.

The result in fig. 4 clearly indicates that most of the participants thought that BST and VIRP were the best techniques. NAT followed by AHP yields less accurate result. CV and PG were located between these two groups. It was expected that AHP would produce the most accurate result as in this method requirements were prioritized according to mathematical rules. An explanation to why AHP more or less did so poorly here can be that the participants did not understand how to read out, the matrix that presented the prioritization results.

e. Number of comparisons: Finally the participants were asked to keep records of how many comparisons were required for each technique. The result is shown in fig. 5.

Figure 5. Comparison on the basis of number of comparisons.

The result in fig. 5 clearly indicates that AHP was required the highest number of comparisons because the number of comparisons in AHP is n(n-1)/2. NAT, VIRP, CV, and PG were required the lowest number of comparisons because they only require n comparisons. BST was in the middle of these two groups, because it require n(log n) comparisons.

V. FINDING THE OVERALL BEST PRIORITIZATION TECHNIQUE

After collecting the data based on above motioned criteria, we assigned weight to each criterion and then applied formula (2) and (3) to find out the best requirements prioritization technique. Each of the above criteria was assigned weight according to Weight Table I.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Ease of use</th>
<th>Total time taken</th>
<th>Scalability</th>
<th>Accuracy</th>
<th>Number of comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>8.5</td>
<td>4</td>
</tr>
</tbody>
</table>

Table I: Weight Table for each criterion
Formulae used to calculate overall score by each of the prioritization techniques under consideration:

\[ C_{ij} = W(C_i)(N+1) - R_i(T_j) \] …………(2)

\[ OS(T_j) = \sum_{i=1}^{NC} \frac{C_{ij}}{NC} \] ………… (3)

Where, N = Number of techniques used
Si,j = Score of technique j in criteria i
W Ci = Weight of criteria i
NC = Number of criteria’s
Ri (Tj) = Ranking of technique j in criteria i
OS(Tj) = Overall score of technique j

The result after calculation is shown in fig. 6

Figure 6. Overall weighted score of prioritization techniques.

Fig. 6 clearly indicates that among all the requirement prioritization techniques under consideration, VIRP is supposed to be the best one based on the mentioned evaluation criteria. This order of the requirement prioritization techniques obtained from this experiment, however, is not a global one as rankings can be reordered if criterion weights are assigned differently. Nevertheless, the technique and formulae used here to compare among different prioritization techniques can be used in any scenario with appropriate criterion weights suitable for that scenario. The generalizability of the paper is limited due to the small sample and the specific context. A real project has requirement’s interdependencies and time and budget pressure to consider, which may cause the decision-making difficult. However, the VIRP technique fits in all the different environments. The main disadvantage of the experiment being the difficulty to generalize to industrial projects, it would be valuable to try the experiment out in a case study. The participating organization would then get knowledge about prioritization and perhaps find a technique that suits them the best.

VI. CONCLUSION

As per our experiment, we can conclude that VIRP is the best technique for requirements prioritization. It is an easy method, it gives the most accurate results, and it is rather comfortable to handle even if there are many more requirements. PG and BST were located in the middle in some questions but they were neither the best nor the worst techniques. However, the test subjects thought that PG was the next-best method of these six techniques to be used when prioritizing requirements. The worst method, according to result is NAT. The reasons for worst performance of NAT are as follows:

- determining the absolute information is difficult than relative information
- participants’ subjective opinions regarding a number differ widely,
- is not effective when numbers of requirements are low, less accurate and informative,
- takes maximum time to prioritize.

However, this order of the requirement prioritization techniques obtained from this experiment is not a global one as rankings can be reordered if criterion weights are assigned differently. Nevertheless, the technique and formulae used here to compare among different prioritization techniques can be used in any scenario with appropriate criterion weights suitable for that scenario.

VII. REFERENCES