Available online at www.sciencedirect.com

ScienceDirect

Materials Today: Proceedings XX (2016) XXX–XXX

www.materialstoday.com/proceedings

PMME 2016

# RFP based Requirement Prioritization – A One-Step Solution

M Soumya Krishnan

*Assistant Professor, Amrita School of Arts & Scineces, Amrita Vishwa Vidhyapeetham, Kochi Campus, Kerala 682024, India*

**Abstract**

Every IT companies start off with their project by prioritizing the RFPs received from their client. These requirement prioritization techniques are practiced to diagnose and categorize the momentous requirements that are put forward by each client. Based on this prioritization strategy, the products are delivered to the client on a timely and needy basis. It is found that, IT companies are following and practicing a number of prioritization techniques as it is considered as a decisive stair. These techniques are chosen purely based on the functional and non-functional requirements of the project indicated in the RFPs supplied by the clients. But the real issue arises when more than one technique get overlapped in many situations. RFPs incorporate both functional and non-functional requirements. A software engineer feels jumbled in prioritizing these requirements. But functional requirement becomes obligatory. So, the only confusion is related with categorizing the non-functional requirements such as cost, dependency, time, risk, benefit, value and effort. The categorization of these requirements becomes dubious and challenging for a software engineer since lot of techniques already is in existence. At this stage, a model can be suggested to avoid confusion in prioritizing the requirements without any misperception. Through this paper, I have tried to put forward and expound a model as a solution to this problem which can be thoughtlessly followed in undergoing the procedure of prioritizing these non-functional requirements in a most simple and effectual manner. This solution model considers all the pre mentioned non-functional elements by providing a perfect way to follow and select a suitable technique to conduct requirement prioritization.


Selection and Peer-review under responsibility of International Conference on Processing of Materials, Minerals and Energy (July 29th – 30th) 2016, Ongole, Andhra Pradesh, India.

*Keywords:* RFP, Requirement Prioritization; Solution Model; Prioritization Techniques; EVOLVE; SERUM; PG; AHP; CV; Cost-value; VOP.

## 1. Introduction

Prioritization of requirement for any software project is one of the foremost and critical steps in the entire software development procedure. Requirement Prioritization can be carried out in two ways – based on its importance and implementation order [5].


Selection and Peer-review under responsibility of International Conference on Processing of Materials, Minerals and Energy (July 29th – 30th) 2016, Ongole, Andhra Pradesh, India.

Importance is based on more than one dimension like individual preference, implementation cost, business value, completion time, business benefits, effort exerted, dependency and risk in development. These candidate non-functional requirements are taken into consideration for my area of study through this paper. Considering and defining these candidate requirements for every project release is a significant task. Thus, one must execute it with at most care while prioritizing each requirement. The success of any project depends on how well the customer requirements are being processed and prioritized. One must be absolutely sure about selecting a prioritization technique as this is considered as the vital constituent for instigating any software project efficiently and as premeditated.

### *1.1. Background Study*

There are many requirements prioritization techniques followed and carried out in every software companies. But deciding on the most appropriate one is the factual problem in its own rights. The typical techniques followed are Planning Game (PG), Analytic Hierarchy Process (AHP), Numerical Assignment Technique (NAT), Cost-Benefit Analysis (CBA), Binary Priority list (BPL), Software Engineering Risk Understanding and Management (SERUM), Cumulative Voting (CV), EVOLVE and Value Oriented Prioritization (VOP). All these techniques have their own pros and cons [3][10]. But picking out the best technique suitable for the stakeholder's requirement gives a chaos to every software analyst. And to resolve such a situation, a simple model has to be followed so that requirement prioritization becomes an easy task for every analyst.

### *1.2. Purpose and Objective*

Doing Prioritization for large complex projects has become a highly convoluted task for every IT firms. And accomplishing a successful output in prioritization has also become an intricate mission. It creates more chaos as there are plentiful techniques concomitant with prioritization and so one can get dazed while selecting any technique for doing so. To find a clarification for this concern, in my initial work, I decided to prioritize a system based on some non-functional factors such as dependency, time, cost, risk, benefit, value and effort. Next by performing a comparative study between these techniques and their features it was clear that no one followed all the stated non-functional requirements [7]. It was found that the reason for such an issues was due to the fact that, no stated process exists to choose these techniques based on the required non-functions. This concern paved an insight and initiated a thought of developing a special solution model to resolve such a situation. Through this paper, my main objective was to put forward a simple model that help simplify the task of business analyst to achieve a better selection of a appropriate and best technique for performing the requirement prioritization based on the RFPs submitted by the clients with in a short time period.

## 2. Literature Review

The study mainly focused on different prioritization techniques which are currently being practiced in software companies. Based on these studies done, development of a new solution model was the main objective of this paper. Thus, the main focus was to given to find out the best technique for doing requirement prioritization. For this many papers were reviewed. The general aspects of Requirement prioritization were observed in paper [1] [2]. These papers gave an overview of how partitions are carried out on requirements and choose the best former among them. Comparisons between various methods were analyzed in [3]. The paper [6] helps in describing the relation among the four major techniques like AHP, SERUM, EVOLVE, VOP and other non-functional aspects like Cost, value, Benefit, Risk, Dependency were described in a beautiful way. Prioritization of requirements - its process, approaches and techniques are neatly explained in [5]. Paper [7] concluded that not any of the techniques followed all the aspects properly. The illustration of how the working of AHP is carried out in [10] with an example which provides a framework in determining the benefit of each requirements that can be used even in immeasurably multifaceted circumstances. The most commonly used technique in agile methodology namely Planning Game was considered in [9][19] which explains how requirements are generated during every release and it also ceased the fact that it scores the best in time consumptions with respect to pair-wise comparison. Some more efficient techniques were derived from [9] that mentioned six techniques like NAT, AHP, VOP, CV, BST, and PG. The conclusion

derived from this paper was that VOP is the best prioritization method since it is an easy way in handling any number of requirements. They also mentioned that both PG and BST neither resulted in a best nor the worst techniques. Paper [12] explains the gradational procedure which demonstrates the process of Cost Benefit analysis which elucidates a straight forward technique on choosing whether to move forward with a project. Along with this, it was found that how Prioritization is carried out based on Cost and Benefit in [13]. The selection of most valuable requirements was clearly depicted in [14] [16] which introduced a VOP framework for determining the business values and the relationship among them for the implementation of successful business decisions. An approach of assigning requirements as increments is explicated in [15] in which EVOLVE explains how to deliver requirements in an evolutionary approach using a genetic algorithm. The paper [17] focuses on the technique to handle prioritization of requirements in an hierarchical way. This method is named as Hierarchical Cumulative voting; an alternative of Cumulative Voting. Risk Management is handled using SERUM in [18] which has implicit and explicit risk management process that is designed to handle the generic risk that is visible throughout the project.

## 3. Theoretical Overview

In this segment, I have tried to summarize the various techniques on requirement prioritization. Although several techniques exist and are under practice, I have chosen some seven main techniques which are extensively used in the industry nowadays. They are as follows:-

### *3.1. Value Oriented Prioritization [VOP]*

The central knowledge behind VOP is to emphasis on the main business values that results in the satisfaction of every stakeholder [16]. This particular technique works out by creating a outline that recognizes business core values and their relationships. It also targets on identifying risks categories and organization's tolerance towards these risks. Using the core business values and risks VOP constructs a prioritization matrix that makes a comparison among them. In conclusion, we can guesstimate the score for each requirement as the sum of its contributions to business values minus sum of its risks.

### *3.2. Analytical Hierarchical Process [AHP]*

AHP offers an operative tool for dealing with the intricate choices by translating into pair wise comparisons [11]. It helps in converting the problems into hierarchy of criteria's and substitutes. It also combines criteria weights and option score to decide on global score for each option and a consequent ranking. Global score for a given option is the weighted sum of scores it obtains with respect to all the criteria.

### *3.3 EVOLVE*

This is prioritization technique which is an evolutionary as well as an iterative approach where software releases are scheduled as increments but the planning process is reiterated [15]. EVOLVE follows the computational strength of genetic algorithm to determine the most optimal plan with the iterative solution method. Here at iteration 'k', a final decision is taken about next increment and allow all kinds of late changes in requirements prioritization. EVOLVE is carried out by using a Palisade's Risk Optimizer Tool. The tool also provides various algorithms for regulating the variables.

### *3.4 Software Engineering Risk Understanding and Management [SERUM]*

SERUM is another prioritization technique that deals with risk management in software development process by using estimation for cost, benefit, development risk and operational risk, reduction in performing the prioritization process [17]. This technique is executed in the following way:

- Modify the proposed system by inspecting the risk in the current system.

- Improve the proposed system by investigating the risk in the proposed system.
- Delineate the changes – all changes will be documented and an evolutionary plan is derived and all the changes are inspected.
- Perform Cost benefit analysis on the proposed changes.
- Prioritize the proposed changes using the cost benefit evaluation.
- Reform the change priority using the risk estimation for the current system, the proposed system and the development process.
- Develop change plan – This activates the implementation phase and helps in the production of developmental schedule.
- Perform risk control plan for identified and accepted risk

### *3.5 Planning Game [PG]*

PG is the main planning process followed in extreme programming (XP). It is used in planning and deciding what is to be developed in XP [8]. The planning process is mainly divided into two parts:

- Release planning – Decides on what all requirements are required to be included in next release.
- Iteration planning – It decides on the main activities and tasks of developers. It has three phases.

These two process is again sub divided into three phases: Exploration Phase, Commitment Phase and Steering Phase

### *3.6 Cumulative Voting [CV]*

Cumulative Voting is a ratio scale technique where a customer partaking in the process is given a fixed number of units for casting their votes [17]. CV is executed against all items of similar level of abstraction. For eliminating this issue we use hierarchical CV where items are first broken into higher level of abstraction and it is further decomposed into detailed level describing features, functions etc. It also provides options for applying multiplier to the votes like most important (multiply by 1) next important (multiply by 0.75) and least important (multiply by 0.50).

### *3.7 Cost Value*

The cost-value approach was aimed to identify and specify requirements for a particular system [6]. Based on the identified high level requirements that enclosed the basics of the system, the project members can review those requirements. Then, the identified requirements can be compared with the customers' interpretations and can be prioritized through pairwise comparisons. The assessments were first conducted according to the requirements' values and then according to the implementation costs. A cost-value diagram was then designed, which provided a clear overview of the requirements that produce high value for relative low costs.

## 4. Formulation of a One-step Simple Solution Model for Requirement Prioritization based on client's RFP

### *4.1. Research Theory & Design*

Software analysts strive a lot in choosing the correct or apt prioritization techniques based on the RFPs received from the client. These RFPs indicate more than one element to be considered in the project to be developed. In such a situation, analysts break their head and follow the customary way of choosing any one or mix of techniques without any primary reason. This approach is practically ineffective and time consuming leading to inept outputs in future. In order to overcome such a situation and get a better practicable solution was my primary objective of research. To certain extent, I could succeed in introducing a simple model which can be used by the analyst as a one-step solution to this problem of prioritization.

The suggested system is a simple transitional process-flow model. It is assured to help the business analysts to make right choice in determining the apt technique for prioritization. The proposed system starts working by

receiving and scanning through the RFPs obtained from the clients. Through this procedure, the analyst can investigate the essential factors mandatory to the project. The next step is to make a comparative study from the Table 1 which indicates the comparison of various non-functional factors and its corresponding prioritization technique. Later, traverse through the suggested model or flow diagram based on conditions derived from Table 1. On following this pathway, it will lead us to the precise, apt and best approach for executing requirement prioritization.

A table depicting various non-functional aspects is indicated as shown in *Table 1*. From this, we are able to evaluate the aspects followed by each prioritization technique. A simple process-flow diagram was formulated using the result derived from the table after performing the analysis. This model articulates a clear-cut flow of procedures that ultimately provides us with a simple one-step solution to perform a pertinent prioritization technique. By following such a process model, it is guaranteed that the preliminary burden of classifying and prioritizing the non-functional requirements of software projects can be resolved to a greater extend.

The proposed solution model for performing the requirement prioritization works as depicted in the *Figure1*.

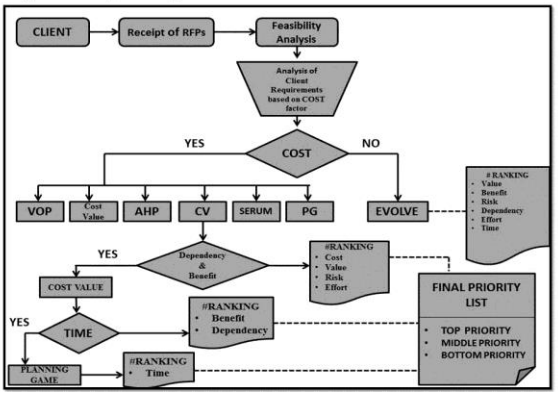
Table 1 -*Comparison of different prioritization techniques based on various non-functional factors*

| Techniques ↓ | Cost | Time | Risk | Value | Effort | Benefit | Dependency |
|---|---|---|---|---|---|---|---|
| **AHP** | Yes | | | Yes | | | |
| **VOP** | Yes | | Yes | Yes | | | |
| **PG** | Yes | Yes | Yes | Yes | Yes | | |
| **Cost Value** | Yes | | | Yes | | Yes | Yes |
| **CV** | Yes | | Yes | Yes | Yes | | |
| **SERUM** | Yes | | Yes | | | Yes | |
| **EVOLVE** | | Yes | Yes | Yes | Yes | Yes | Yes |

Immediately after the receipt of an RFP from the client, the usual feasibility analysis will be performed. After that, based on several non-functional requirement factors, cost will be considered primarily. Now, based on the cost factor the model depicts two path ways. The business Analyst can now blindly choose EVOLVE if cost is not a constraint or cane traverse through the other direction when cost is included. This helps the analyst to minimize the time span and complication associated with the most important segment of software project development. Once traversing through the cost based pathway, one can move forward depending upon other aspects as shown in *Figure1*. In this wing, other options for choosing techniques are clearly depicted. These techniques use their own priority checking tools and rank each factors according to their importance in the project. Each technique finally gives a prioritized list having three layers of priority -. *TOP*, *MIDDLE* and *BOTTOM* Priority. These are prepared based on the ranking given to each requirement. This prioritized list may vary based on the projects size, complexity and type as indicated in the RFPs submitted by the client.

Ultimately, the solution model as shown in *Figure-1* gives us a one-step simple approach in deciding on the techniques to perform the requirement prioritization based on RFPs supplied by the clients. This flow model helps every software analyst to decide what to do and how to move forward to do prioritization in a single glance at the flow diagram.

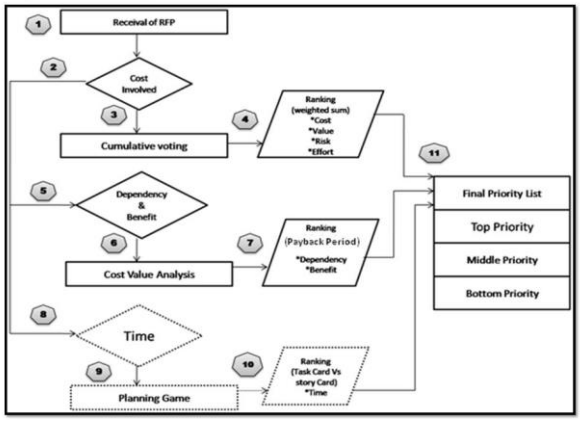
Figure 1: - Proposed Requirement Prioritization Solution Model



*4.2. Functioning of the proposed Solution Model - A Case Study*

The above figure (*Figure-1*) clearly depicts the work flow of the suggested solution model for performing the requirement prioritization based on the non-functional requirements indicated in the RFPs submitted by the clients. The working of the selection model is evidently illustrated by taking a sample RFP – City of Duncan (COD) Website Redevelopment [4]. The RFP demands for the implementation of an already existing software model with additional functionalities. The main goal of this RFP is for improving access to online information and also for providing services for businesses, citizens and viewers external to the community and staff. This project mainly focuses on the non-functional aspect - *value*. COD wants to develop the project in a web platform and they also need the exact version in mobile platform. This makes the implementation more complex. If this is considered from the developer's point of view, the *effort* and the *risk* related with the development the project is critical. Taking into consideration the above criteria, the selection procedure is portrayed in Figure-2. If this project requires completion within a particular *time* span, then the path extends by performing an additional technique named Planning Game which is shown as doted lines in Figure-2.

Figure 2 – Functioning of the Solution Model based on "COD" RFP




The functioning of the suggested Solution Model is illustrated in a step wise manner as follows:-

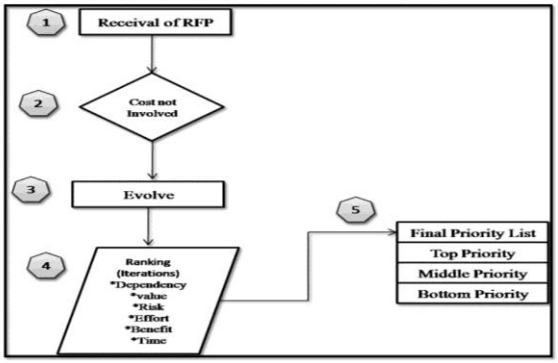
1. RFP receipt from the client.
2. Finding whether cost is an important criterion. Cost is considered as an important criterion for above RFP.
3. ***CV*** is considered best technique here since it's a complex project considering *risk* factor.
4. Ranking based on the weighted sum is performed.
5. From the RFP it is clear that project *values* more on *benefit* than *cost* and also the project is to redesign from an existing one, hence *dependency* should also be considered.
6. Taking both *dependency* and *benefit* as a vital factor, ***Cost Value*** is the best technique to consider.
7. Next, Ranking based on Cost Value is done (by using the payback period).
8. If the project strictly specifies the time of completion, then *time* factor needs to be considered.
9. Taking time into consideration P***lanning Game*** is the best technique for prioritization.
10. Ranking based on planning game is done by making a comparison with the task card generated by the programmers and story card generated by the customer.
11. Final priority list is derived from the above three rankings.

Now, let us consider another instance where cost is not measured as a prime factor. Then, one may have to traverse through the right wing of Figure 1. The working of such a situation is depicted using *Figure-3.* The work flow of *Figure-3* can be illustrated as follows:-

1. RFP receipt from the client.
2. Finding whether *cost* is an important criteria .The RFP "COD" does not focus on *cost.*
3. If the project focuses on all the factors except that of *cost*, ***Evolve*** is the best technique.
4. Ranking based on Evolve through an iterative approach is performed.

5. Final priority list is derived from the above ranking.

Figure -3 Functioning of the Solution Model when C*ost* factor is not considered




By using the sample RFP of "COD" and by illustrating the different conditions, the suggested simple Solution Model working is best described here in this paper. Now, by following this method, any software analyst can now confidently use this solution model to perform the requirement prioritization without any confusion.

## 5. Conclusion

Requirement Engineering is a complicated phase that involves various tough and monotonous activities in the process of a software development. Within requirement phase, the most intricate task is to prioritize the requirements which contain both functional and non-functional elements. Prioritizing functional aspects is much more systematic compared to non-functional requirements in a technical point of view. But performing prioritization for large complex projects has become a highly complicated task for every IT firms nowadays. And this confusion gave rise to a research idea of developing a simple solution model which helps to resolve such an issue to a greater extend by saving time and effort of a software analyst. The motivation of this research paper was to minimize these complications and help one prioritize requirements agilely and effectively. The practice utilized here is a systematic literature review which helped to identify the most prominent techniques along with their key features of requirement prioritization. As a concluding note, it is evident from the studies and research done here, a simple solution model was developed and proposed, which could assure the selection of suitable and best prioritization technique based on any RFPs submitted by the client.

## Acknowledgements

I would like to extend by heartfelt gratitude to Mr. Mahesh Bala, CEO, OptioLogic Technologies Pvt Ltd, for supporting and providing me with sufficient knowledge about how requirement prioritization is carried out in software firms. I am also immensely grateful to the employees of OLT for their full support in providing me with data and resources to perform my study in a successful way. I would also like to extend my deepest gratitude to our Director, Dr. U. Krishnakumar for his continuous motivation and support in doing this paper. Last but not the least; I would like to thank my MCA students for their help in providing me with essential information necessary for this study.

## References

[1].Achimugu Philip, Selamat Ali, Ibrahim Roliana, Mahrin Naz'ri., A systematic literature review of software requirements prioritization research, Information and Software Technology, (2014), Vol. 56, No.568-585.

[2]. Barros Márcio, Pitangueira Mauricio. , Maciel P. Suzana., Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the Literature, The Journal of Systems and Software, (2015), Vol.103, No.267-280.

[3].Babar Imran, Mohammad Radziah, Jawawi N.A., Falaksher, Requirements Prioritization Techniques and Different Aspects for Prioritization, 8th Malaysian Software Engineering Conference (MySEC), (2014) , 978-1-4799-5439-1.

[4]. City of Duncan Website Redevelopment http://www.duncan.ca/pdf/2015-08 04%20City%20Website%20RFP.pdf.
[5]. Firesmith Donald, Prioritizing Requirements, Journal of Object Technology, ETH Zurich, Chair of Software Engineering, (2004), Vol. 3, No.8,
[6]. Karlsson J, Ryan K, A Cost-Value Approach for Prioritizing Requirements, IEEE Software 14(5):67-74, (1997).
[7]. Iqbal Aaquib, Khan M., Khan A, A critical analysis of techniques for requirement prioritization and open research issues, International Journal of Reviews in Computing, (2009), Vol.1.
[8]. Karlsson Joachim, Wohlin Claes, Regnell Bjorn, An evaluation of methods for prioritizing software requirements, Information and Software Technology, (1998) , Vol.39, No.939-947.
[9]. Karlsson Lena, Thelin Thomas, Regnell Bjorn, Berander Patrik, Wohlin Claes , Pair-wise comparisons versus planning game Partitioning—experiments on requirements Prioritization techniques, Empir Software Eng, (2007) , No.12:3–33.
[10]. Khari Manju and Kumar Nikunj, Comparison of six prioritization techniques for software requirements, Journal of Global Research in Computer Science, (2013), Volume 4, No. 1.
[11]. Dr .Meixner Oliver, Dr. Haas Rainer, An Illustrated Guide to the ANALYTIC HIERARCHY PROCESS, Institute of Marketing & Innovation University of Natural Resources and Applied Life Sciences, Vienna (http://www.boku.ac.at/mi/), (2014).
[12]. Mind Tools" Cost-Benefit Analysis Deciding, Quantitatively, Whether to go Ahead (Also known as CBA and Benefit-Cost Analysis), (https://www.mindtools.com/pages/article/newTED_08.htm).
[13]. Nickel Julia, Ross M., Rhodes H., Comparison of Project Evaluation Using Cost-Benefit Analysis and multi-attribute Trade space Exploration in the Transportation Domain , Second International Symposium on Engineering Systems MIT, Cambridge, Massachusetts. (2009)
[14]. Padmanabhuni Srivinas, Boehm Barry, Payyavula Swaroop., Kukreja Nupul, Value-Based Requirements Prioritization: Usage Experiences Conference on Systems Engineering Research (CSER'13) Procedia Computer Science, (2013), Vol.16 No. 806 – 813.
[15]. Ruhe G. , Greer D, Software release planning: an evolutionary and iterative approach, Information and Software Technology, (2004), Vol. 46, No.243–253.
[16]. Smith K., Azar Jim, CordesDavid, Value-Oriented Requirements Prioritization in a Small Development Organization, IEEE SOFTWARE Published by the IEEE Computer Society, (2007).
[17]. Svahnberg Mikael, Berander Patrik, Evaluating two ways of calculating priorities in requirements hierarchies-An experiment on hierarchical cumulative voting, The Journal of Systems and Software, (2009), Vol. 82, No. 836-850.
[18]. W. Bustard. , Greer D, SERUM - Software Engineering Risk: Understanding and Management, Faculty of Informatics, University of Ulster, Cromore Road, Coleraine, BT52 1SA, Northern Ireland.
[19]. Wikipedia, the free encyclopedia, Extreme programming practices.