**I J C T A**, 9(15), 2016, pp. 7489-7496
© International Science Press

# Solution Model for Requirement Prioritization

**Aswani Viswanathan[1], Smrithi R. Nair[2] and M. Soumya Krishnan[3]**

**ABSTRACT**

Requirement Prioritization strategies are practiced in software project development to determine and manage the significant requirements that are needed to be delivered in each stages of a software release. For handling requirement prioritization we have different techniques, where we choose the appropriate technique based on the customer's requirements. But choosing the appropriate technique may encounter problems as two or more technique can overlap each other. So to determine the most efficient technique certain factors like cost, dependency, time, risk, benefit, value and effort was incorporated to build a process model which can tackle and minimize the problems associated with prioritizing the requirements. Systematic review of this model is to provide an easy path to determine the perfect prioritization technique based on different non-functional factors. This Process-model considers all the above non-functional factors by providing a pathway in selecting the suitable technique.

***Keywords:*** Requirement Prioritization, Techniques, non-functional factors, Process flow model, RFPs, PG, AHP, EVOLVE, SERUM, CV, CBA, SERUM,

## 1. INTRODUCTION

Requirement Prioritization is used in software project development to determine which all requirements needs to be delivered in each stages of a software release [1].So that, all the candidate requirements are well understood and the process to execute them can be carried out quickly. Determining these candidate requirements for every release is an important task, so one must give at most caring 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 key element for implementing any software project efficiently and as planned.

### 1.1. Background

Requirement Prioritization is carried out using several techniques. The techniques considered were Binary Priority List, Planning Game, Cost-Benefit Analysis, Analytic Hierarchy Process, Numerical Assignment Technique, Cumulative Voting, Value Oriented Prioritization, Software Engineering Risk Understanding and Management (SERUM) and EVOLVE. All these techniques referred only to their pros and cons [3]. As a result deciding on which prioritization technique to use for a specified software development put the analysts on doubt.

### 1.2. Motivation

For project producing large complex software, prioritization becomes a more intricate task. Hence, achieving success in complex projects become a tiring mission. As there are numerous techniques associated with prioritization one can get bewildered while choosing any technique for scheduling. To address this issue, in our initial work, we decided to prioritize a system based on several non-functional factors such as cost,

[1,2,3] Department of CS & IT Amrita School of Arts and Sciences Kochi, Amrita Vishwa Vidyapeetham, India, *Emails: aswaniviswanathan399@gmail.com, Smrithirn1993@gmail.com, mail2sou@yahoo.com*

dependency, time, risk, benefit, value and effort. By making a comparative study between these techniques and their factors it was clear that none followed all the mentioned non-functions [5]. From this we initiated a thought of developing a Process flow model.

### 1.3. Objective

The difficulties faced by the program analyst while selecting an appropriate predominant method for a project results in time complexity and thus affects the process of software development from the very beginning to its closing stage. This paper aims to simplify the task of system analyst by introducing a Process flow model that will help him to perform a better selection of an appropriate and best technique for requirement prioritization based on RFPs provided by the customers in concise time span.

## 2. LITERATURE REVIEW

The general aspects of Requirement prioritization were examined [1] [2]. It describes in more detail how partitions are carried out on requirements and select the most prior among them.

Comparisons between different approaches were analyzed [3]. A paper describing the relation among the four major techniques (AHP, SERUM, EVOLVE, VOP) and factors like Cost, value, Benefit, Risk, Dependency were incorporated to it [5]. They concluded that none of the techniques followed all the factors discussed. From the exploratory study [6] reveals the concept of the oldest method AHP and Hierarchy AHP. The illustration of how the working of AHP is carried out [9] with an example which provides a framework in determining the benefit of each requirements that can be used even in immensely complex situations.

The most commonly used technique in agile methodology namely Planning Game was considered [7][17] 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 [8] that mentioned six techniques (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 cited that both PG and BST neither resulted in a best nor the worst techniques.

A gradational procedure which demonstrate the process of Cost Benefit analysis [10] which elucidate a straight forward technique on choosing whether to proceed a project. Parallel to this we cornerstone how Prioritization is carried out based on Cost and Benefit [11]. They conducted a systematic review to categorize requirements as Primary and Secondary and the technique will be more efficient if it gets validated with the help of organization and other stakeholders. The selection of most valuable requirements was the content of [12] [14] which introduced a VOP framework for determining the business values and the relationship among them for the implementation of successful business decisions.

A method of allotting requirements as increments [13] called EVOLVE that deliver requirements in an evolutionary approach using a genetic algorithm. It can accommodate late changes and the solutions may or may not be optimal.

For handling prioritization of requirements in hierarchies [15] where large requirements are handled in different levels of abstraction so that all detailed level requirements are considered for ordering. This approach is termed as Hierarchical Cumulative voting; a variant of Cumulative voting.

Risk Management is handled using SERUM [16] that has implicit and explicit risk management, so that the development process has designed to handle the generic risk that is visible throughout the project.

## 3. THEORETICAL BACKGROUNDS

In this work, we considered different techniques and their comparison with various non-functional factors. The following Sub-sections describe them in detail.

### 3.1. Planning Game [PG]

The main planning process in extreme programming (XP). It is used in planning and deciding what is to be developed in XP [7]. The planning process is mainly divided into two parts:

(1) Release planning – What all requirements needed to be included in next release.

(2) Iteration planning – Activities and tasks of developers. It has three phases.

The above two process is again divided into three phases : Exploration Phase, Commitment Phase and Steering Phase

### 3.2. Analytical Hierarchical Process [AHP]

It provides an effective tool for dealing with the complex decisions by converting into pair wise comparisons [9]. Here, problems are decomposed into hierarchy of criteria's and alternatives. AHP combines criteria weights and option score to determine 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. Cost Benefit Analysis [CBA]

It is an analytical tool for determining pros and cons of carrying a business proposal [10]. It focuses on adding benefits and then comparing with the cost associated with it and the results are expressed as payback period.

$$\boxed{\text{Total Cost / Total Benefit = Payback Period}}$$

### 3.4. EVOLVE

It is an evolutionary as well as an iterative approach where software releases are planned as increments but the planning process is repeated during every iteration [13]. 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 using a Palisade's Risk Optimizer Tool. The tool also provides different algorithms for adjusting the variables.

### 3.5. Value Oriented Prioritization [VOP]

Main idea behind VOP is to focus on core business values that results in stakeholder's satisfaction [14]. VOP is carried out by establishing a framework that identifies 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. Finally, we can estimate the score for each requirement as the sum of its contributions to business values minus sum of its risks.

### 3.6. Cumulative Voting [CV]

CV is ratio scale techniques were a stakeholder participating in the process is given fixed number of units for casting their votes [15]. CV is done against all items of same level of abstraction. For addressing this issue we use hierarchical CV were 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. Software Engineering Risk Understanding and Management [SERUM]

SERUM is a technique that handles risk management in a software development process by using estimation for cost, benefit, development risk and operational risk, reduction in performing the prioritization process [16]. Technique is performed in the following ways:

- Reform the proposed system by examining risk in the current system.
- Reform the proposed system by examining the risk in the proposed system.
- Define changes – changes will be documented and derive an evolutionary plan were all changes are examined.
- Carry out 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 – Activates for implementation phase and production of developmental schedule.
- Perform risk control plan for accepted risk

## 4. RESEARCH THEORY

The proposed system is an intermediate process-flow model. This will help the system analysts, to make right choice in deciding the apt technique for prioritization. The technique is carried out based on the RFPs produced by the customers from whom we analyze the crucial factors required by the project. We then make a comparative study from the table and traverse through the model based on conditions derived. This pathway will lead us to the correct, suitable and best approach for prioritization

### 4.1. Design

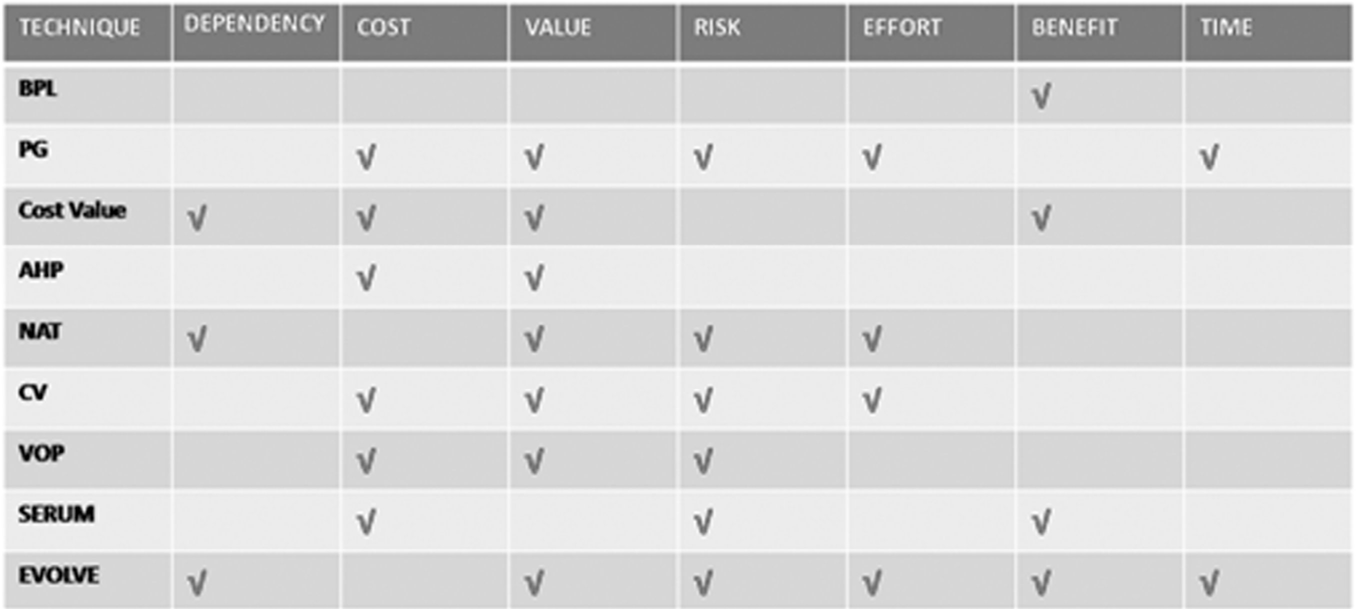
The analysis of the existing techniques leads to a creation of a table categorizing various non-functional factors. From this table [Table: 1] we are able to analyze the factors followed by each technique. A process-flow diagram was resulted using the evidences got through the analysis from the table. This model expresses a clear-cut flow of methods that finally ends up in the felicitous Prioritization technique.

The proposed system [Fig. 1] is to derive a model that depicts all the factors easily by providing two pathways either by choosing EVOLVE if cost is not a constraint or the other if cost is included, thereby reducing the time span and complexity associated with the most important phase of software project development. These techniques use their own priority checking tools and rank each factors according to their importance in the project. Finally, a prioritized list having 3 layers i.e. TOP Priority, MIDDLE Priority and BOTTOM Priority is prepared based on the ranking given to each requirement. This Prioritized List may vary based on the projects developed.

### 4.2. Case study

The working of the above model was depicted using RFP, City of Duncan (COD) Website Redevelopment [4]. The RFP demands for the implementation of an already existing software model with additional

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



| TECHNIQUE | DEPENDENCY | COST | VALUE | RISK | EFFORT | BENEFIT | TIME |
|---|---|---|---|---|---|---|---|
| BPL | | | | | | √ | |
| PG | | √ | √ | √ | √ | | √ |
| Cost Value | √ | √ | √ | | | √ | |
| AHP | | √ | √ | | | | |
| NAT | √ | | √ | √ | √ | | |
| CV | | √ | √ | √ | √ | | |
| VOP | | √ | √ | √ | | | |
| SERUM | | √ | | √ | | √ | |
| EVOLVE | √ | | √ | √ | √ | √ | √ |

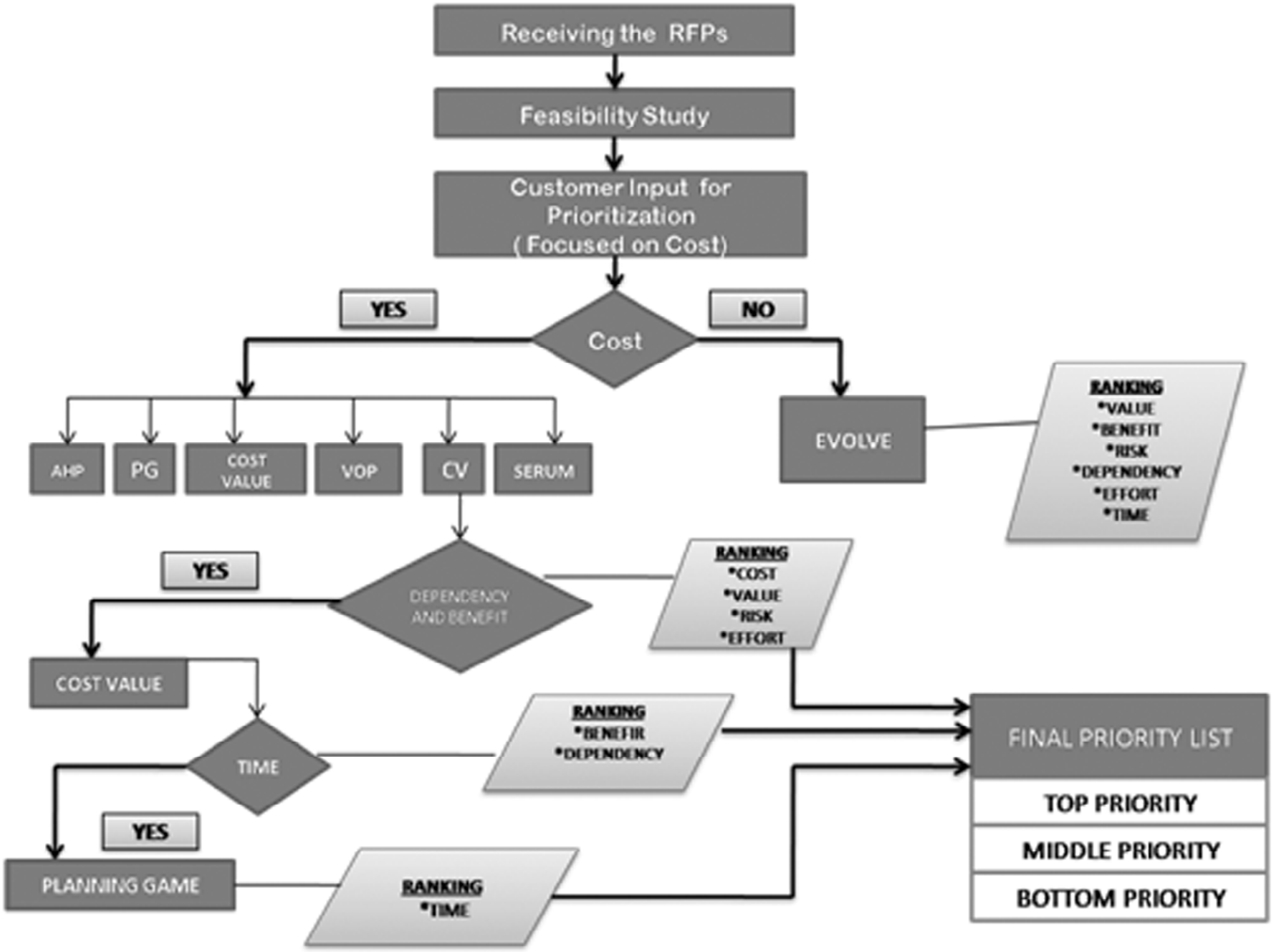


**Figure 1: Proposed Prioritization Process-Flow Model**

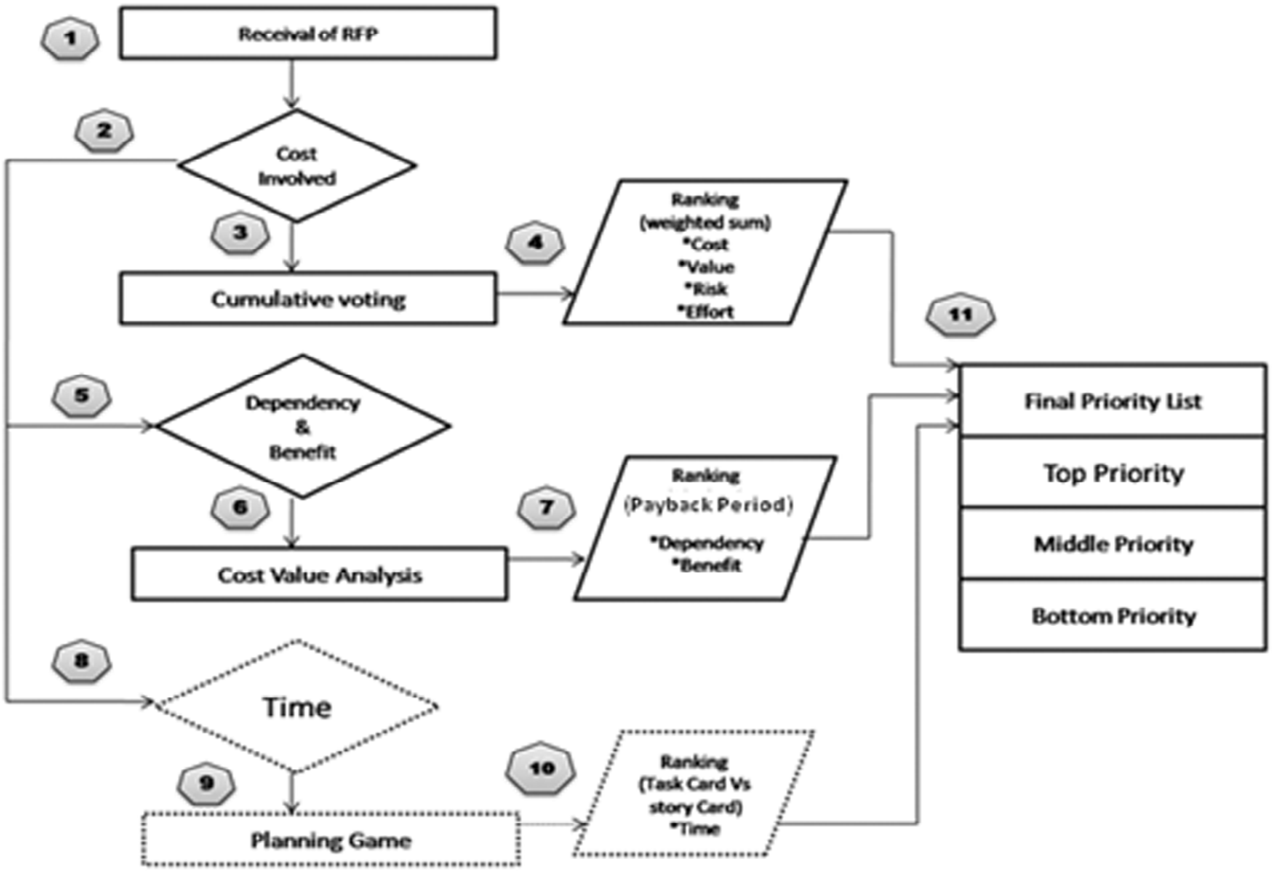


**Figure 2: Working of COD RFP**

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 value. COD desire to develop in a web platform and they need the exact version in mobile platforms. This makes the implementation more complex. If this is considered from the developer's point of view, the risk and the effort associated in developing the project is crucial. Have considered the above criteria, the pathway is depicted in Fig. 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 illustrated as doted lines in Fig: 2.

The working of the above depicted model is as follows:

1. Receiving of RFP's
2. Checks whether cost is an important criteria. The above RFP focuses on cost
3. Since this is being a complex project, developmental risk needs to be considered for which CV is the best technique.
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 redesigned from an existing one hence dependency is considered.
6. Taking both dependency and benefit as an important 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 planning 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 3 rankings.

If we encounter circumstances where cost is not considered a prime factor then, this is the path followed [depicted in Fig: 3].

1. Receiving of RFP's
2. Checks whether cost is an important criteria .The above RFP does not focus on cost
3. If the project focuses on all the factors except that of cost, Evolve is the suitable technique.
4. Ranking based on evolve through an iterative approach is carried out.
5. Final priority list is derived from the above ranking.

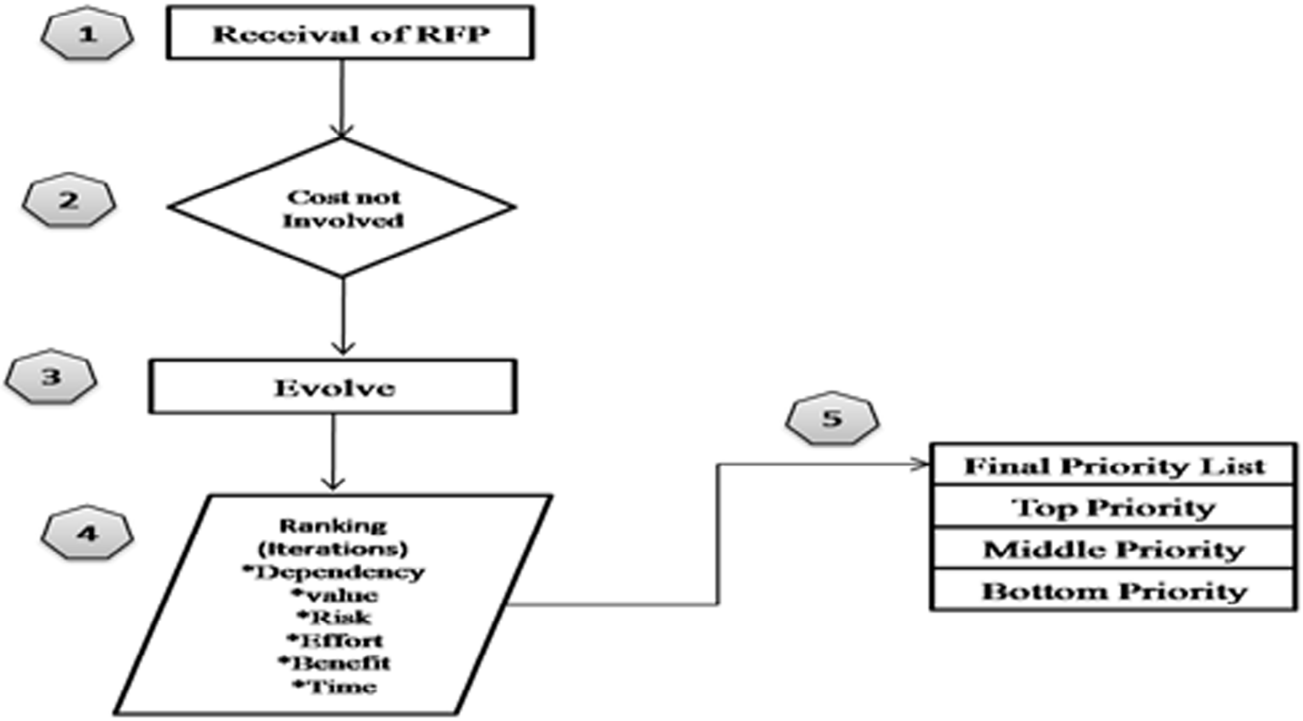


**Figure 3: Working of the model if Cost is not involved.**

## 5. CONCLUSION

Requirement Engineering is an intricate task that involves numerous tedious activities in the process of a software development. Some major dominant mission encountered is the selection and prioritization of requirement which one may find exhaustive and strenuous. The intent of this research is to minimize these difficulties and help one prioritize requirements nimbly and effectively.

The practice utilized here is a systematic literature review which helped us identify the most prominent techniques along with their key features of requirement prioritization. Without a direct indication on the given requirements, a system was developed that uses certain non-functional aspects to prioritize these conditions using the Process- flow model.

## REFERENCES

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

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

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

[4] City of Duncan Website Redevelopment *http://www.duncan.ca/pdf/2015-08 04%20City%20Website%20RFP.pdf.*

[5] Iqbal Aaquib, Khan M., Khan A. (2009) *"A critical analysis of techniques for requirement prioritization and open research issues "*, International Journal of Reviews in Computing, Vol.1.

[6] Karlsson Joachim, Wohlin Claes, Regnell Bjorn (1998) *"An evaluation of methods for prioritizing software requirements"*, Information and Software Technology, Vol.39, No.939-947.

[7] Karlsson Lena, Thelin Thomas, Regnell Bjorn, Berander Patrik, Wohlin Claes (2007) *"Pair-wise comparisons versus planning game Partitioning—experiments on requirements Prioritization techniques",*Empir Software Eng, No.12:3–33.

[8] Khari Manju and Kumar Nikunj (2013) *"Comparison of six prioritization techniques for software requirements"*, Journal of Global Research in Computer Science Volume 4, No. 1.

[9] Dr .Meixner Oliver ,Dr . Haas Rainer (2014) *"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/).

[10] 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).

[11] Nickel Julia, Ross M. ,Rhodes H. (2009) *"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.

[12] Padmanabhuni Srivinas , Boehm Barry ,Payyavula Swaroop . , Kukreja Nupul (2013) *"Value-Based Requirements Prioritization: Usage Experiences* "Conference on Systems Engineering Research (CSER'13) Procedia Computer Science, Vol.16 No. 806 – 813.

[13] Ruhe G. , Greer D (2004) *"Software release planning: an evolutionary and iterative approach"* Information and Software Technology Vol. 46, No.243–253.

[14] Smith K., Azar Jim, CordesDavid (2007) *"Value-Oriented Requirements Prioritization in a Small Development Organization"* IEEE SOFTWARE Published by the IEEE Computer Society.

[15] Svahnberg Mikael, Berander Patrik (2009) *"Evaluating two ways of calculating priorities in requirements hierarchies-An experiment on hierarchical cumulative voting"* The Journal of Systems and Software Vol. 82, No. 836-850.

[16] W. Bustard. , Greer *D "SERUM - Software Engineering Risk: Understanding and Management"* Faculty of Informatics, University of Ulster, Cromore Road, Coleraine, BT52 1SA, Northern Ireland.

[17] Wikipedia, the free encyclopedia *"Extreme programming practices"*.