**Circuits and Systems, 2016, 7, 1008-1020**
Published Online May 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.76085

# Multi-Criteria Decision Making Using ELECTRE

**S. A. Sahaaya Arul Mary[1], G. Suganya[2]**

[1]Jayaram College of Engineering and Technology, Thuraiyur, India
[2]VIT University, Chennai, India
Email: suganyakaruna@gmail.com

Received 23 March 2016; accepted 22 May 2016; published 27 May 2016

## Abstract

**Requirements prioritization is one of the key factors in deciding the success of the project and hence the software industry. One of the major concerns in software prioritization techniques is that the existing ranking techniques have a very modest support to different criteria used by stakeholders to present their ranking. The current techniques are not suitable for arriving at an optimized view of multiple stakeholders using multiple criteria. This research analyzes the issues in existing techniques. A web based decision support model using ELECTRE as the method for prioritization is proposed. ELECTRE is a multi-criteria decision making model that is proved to be effective in ranking several decision making problems. The proposed system takes input from multiple stakeholders using 100-point method. An optimized ranking is obtained using ELECTRE method. The developed system is validated using a pilot project and is found to be efficient in terms of saving cost of implementation and man-hours needed for implementation.**

## Keywords

**Requirements Prioritization, ELECTRE, Ranking, Decision Support System**

## 1. Introduction

The quality of a software product delivered to the customer is frequently determined by the ability to satisfy the needs of the customers and users [1]. Quality can be defined as the degree to which a system, component, or process meets customer or user's needs or expectations. Thus the delivered product should be traceable to the requirements given by the customer. Hence, requirements elicitation is a major step towards the success of any project. If wrong requirements are implemented and users start resisting using the product, it does not matter how firm the product is or how thoroughly it has been tested. The entire effort afforded for the development will go waste.

Traditional way of creating software products starts with requirements analysis followed by high and low level designs, implementation and testing. These steps found to be useless and annoying in today's fastest world. As a result, several development methodologies have been proposed by practitioners to accommodate the needs of industry. To cope up with the ever-changing needs of the industry, software industry is moving towards agility from the traditional approach taking into account the advantages being fast and when delivery of products is not in full but in pieces to customer. Through the concept of having multiple working iterations, the implementation of agile methods allows the creation of quality, functional software with small teams and limited resources.

Requirements Engineering (RE) is the process of establishing the services that the customer requires from a system within the specified constraints of operation and development. A high quality Requirements Engineering can produce a quality product [2]. The following quote from Fredrick Brook's illustrates why requirements are so important: "The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all of the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later". In the traditional approach, Requirements Engineering includes various activities like elicitation, analysis, documentation, validation and management. RE starts and comes to an end in the first phase of project development itself. Agile takes a different approach by allowing RE during the development of product thereby allowing the changes in the development.

In RE, Requirements prioritization seems to be a very major activity and it can be defined as the process of selecting the most important requirement at the particular time and this can be affected by so many factors like cost and budget constraints, availability of personnel, logical implementation order, importance to customers, negative effect, etc. [3]. Several researchers [4]-[6] have analyzed the above role of requirements prioritization in software development methods. Ruhe *et al.* summarize the importance of requirements prioritization as: "The challenge is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized" [6]. This attitude toward requirements makes cost, effort estimation and software architecture development more difficult. But it makes verification easier than traditional methods. Without knowing the final form of the product, or marketplace demands, estimation seems to be impossible [7]. Requirements prioritization in agile environment seems to be even more difficult due to so many factors like changing requirements during the course of software development, the constraint that the piece of product once delivered should not be altered during the subsequent release of other pieces etc. The way of handling requirements seems to be different among companies that range from small scale to enterprises. Most importantly, the meaning of user-specified requirements is clearly known to the developers only during the development of the project. Taking into account all these criteria, RP seems not to be a onetime activity but has to be continued during the course of the project.

Various prioritization techniques are suggested and used to prioritize the most important requirements; and the process is still immature [7]. Those methods prioritize on the basis of factors such as cost, time, and relevant importance. Since the aspects are dependent on each other, existing approaches cause conflicts. These conflicts are initiated due to the effect of one aspect on another aspect. The proposed approach in this paper reduces the conflict by providing thresholds for each stakeholder. The method optimizes the view of multiple stakeholders through the use of arriving at concordance for each requirement. The method is found to be useful and can be completely automated which makes the ability of collecting requirements from geographically distributed stakeholders easier.

This paper has been organized as follows. Sections 2 deals with the background work to be performed for prioritization. A detailed analysis of existing techniques is discussed in Section 3. In Section 4, we have discussed about the description of chosen method ELECTRE. Section 5 deals with the proposed prioritization approach. The case study with a detailed numerical example is given in Section 6. Sections 7 and 8 present the comparison between traditional method and proposed method.

## 2. Background

### 2.1. People Involved in Prioritization

The concept of prioritization is backed up by so many people called stakeholders. Number of stakeholders involved in the project may range from one to many. Even if the stakeholder is one, in practical it will not be so. For example, the owner and user of a product may be different. Usually, stakeholders may range from customers, developers, project managers etc. Every stakeholder involved in the prioritization process will have their own view and they don't want to get compromised on others views. Customers used to prioritize the requirements according to the perspective of how valuable it is to them. And also Individual customer's decision upon the prioritization of requirements seems to be inefficient in making decisions.

Developers used to prioritize the requirements according to the cost, risk involved and other criteria associated with a specific requirement. Developers also concern more about the impact of requirements on the product architecture and hence decide the priority. A project manager needs to balance about the scope of the project aligned with the schedule, budget, staff availability, quality of the product etc., while setting priorities, manager has to balance the benefit that each requirement provides against its cost and the penalty it has for the product's architectural foundation for future evolution. The importance of different stakeholders involving in a project has been studied by so many researchers [8]-[11]. The studies show that the quality of final product is well accepted if proper stakeholders are being involved right from the start of the product development.

### 2.2. Impact of Prioritization on Small and Medium Scale Firms

Small and Medium firms are often considered "motors" of industrial growth: They are very dynamic, innovative and efficient. In a survey by Uolevi Nikula, Jorma Sajaniemi and Heikki Kälviäinen [11] with twelve software companies, it was found that the RE practices in practice lack automation and hence need improvement. In an article about India's Software industry [12], Subash presented the pyramidal structure followed by India's software industry. In that article he was insisting that small and medium firms play a significant role in domestic market. Jim Azar *et al.* [13] insisted the importance of requirements prioritization in small scale firms. He insisted that in small scale companies, proper requirements selection means a large about the survivability of the company. According to an industry study, 53% of project prioritization is driven by the politics within the organization. In order to cope up with these kinds of situations, automatic prioritization software's are particularly necessary for success and survival of the company.

## 3. Business Considerations in Prioritization

Requirements prioritization can be done by taking the attributes of the requirement into account. An attribute or a factor of a requirement is the property of it. Various authors refer to this factor by various names like aspect, criteria, element etc. Several factors that accounts for prioritization are importance, penalty, cost, time, volatility, risk, resources, market strategies etc. Determining priority among the requirements depending upon one factor seems to be easier. But in practice, prioritization needs to be done by taking more than one factor into account. Variation in one factor certainly affects other factor. The list of factors to be considered depends upon the situation of development. Some of the important factors that should be considered for prioritization are listed below.

### 3.1. Importance

Importance of a requirement refers to the weight of corresponding requirement of the system under development. The meaning of importance depends on the viewpoint of the stakeholder. It may be the urgency of implementation due to market competitions, importance concerned to the product under development, importance concerned to the company etc.,

### 3.2. Penalty

Penalty refers to the price the company has to loss when the requirement is not satisfied and it can be taken as a complement of importance of a requirement. While calculating penalty, the rate of how unhappy the customers will be if the feature is not present needs to be calculated. Penalty also includes the rate to be suffered by the developer when the requirement is satisfied late in the development process.

### 3.3. Cost

Cost is a major factor in prioritizing requirements and it is usually calculated by the organization. Most organizations consider cost purely in terms of money, but this may also be calculated in other terms such as number of man hours used for creation of a requirement etc., Many factors correspond to the estimation of cost and it includes the complexity in the development of a requirement, whether or not the code can be reused, the documentation models to be produced, availability of technical personnel etc.,

### 3.4. Risk

Risks associated with each and every requirement plays a vital role. Risk includes both internal risks that exists within the organization (for e.g., technical and market risks) and external risks (regulations given by the regulatory bodies, survival of customers etc) . People may attempt to implement the requirements having the highest risk first so as to deal with the resulting problems during development. Alternatively, it may make sense to implement the lowest risk requirements first in order to maximize the amount of the system implemented by ensuring that limited resources are not wasted on trying to implement high risk aspects of the system that may be impossible for successful completion. Postponing the implementation of high risk requirements can also maximize the time available to analyze about the risks and determine appropriate risk easing approaches.

### 3.5. Time

Time refers to the lead time which starts when the request is received from the customer and ends at delivery of product to customer. This is actually the customer sees. Prioritization helps in reducing the lead time thereby increasing several factors like increasing the trust with the customer, increasing sales etc.

### 3.6. Volatility

Requirements volatility represents the tendency of requirements to change over time. Unstable requirements affect the stability and planning of a project and probably increase the costs since changes during development increase the cost of a project. The rate of volatility is measured as a ratio of the total number of requirements changes (add, delete, and modify) to the total number of requirements in the system over a period of time.

### 3.7. Availability of Resources

Resources refer to the budget, staff and schedule. Resource estimation is one of the crucial factors in requirement prioritization, e.g., instead of waiting for a developer, the requirement that can be satisfied by the existing developer can be fulfilled which may reduce the lead time.

### 3.8. Dependency

Certain requirements depend on other requirements. The dependency among different requirements can be known by drawing use-case diagram for the given list of requirements. Dependency plays a major role in prioritization the delivered piece of the product should not be altered at any time.

### 3.9. Changing Priorities

The priorities of requirements will change during the course of product development. Some of the reasons for changing requirements are:

- Customers tend to prioritize requirements initially from the perspective of how valuable the functionality is to them. After having the discussion with developer, designer etc., about the cost, technical risk, or other trade-offs associated with a specific requirement, though, the customers might decide it isn't as essential as they thought earlier.
- The business environment and needs may change.
- The developers might also determine that certain functions which have low priority should be implemented early on because of their impact on the product architecture [14].
- In environments where agile methodologies are followed, some requirements may essentially need to be implemented before others since agile does not encourage change in delivered pieces of software.
- There may be change in stakeholders due to movement of people.
- Requirements with respect to individuals may change.

## 4. Related Works

Racheva *et al.* [15] presented a survey by assessing a number of requirements prioritization techniques. Based on the descriptions suggested by them, these techniques can be classified into two main categories: techniques that can be used to prioritize small amounts of requirements (small-scale) and techniques that scale up to very large scale (medium-scale or large-scale), thus can be used for the prioritization of larger amounts of requirements. Small-scale techniques can usually be used without the aid of a software tool and are often relatively simply structured.

The techniques mentioned by Racheva *et al.* [15] are the round-the-group prioritization, the $100 allocation technique, the multi-voting system, the pair-wise analysis, the weighted criteria analysis, the dot voting technique, and the Quality Functional Deployment approach. Thomas Bebensee *et al.* discussed about the use of Binary Priority List to prioritize the requirements. The technique works well when the number of decision maker is one. It is not able to satisfy situations when the number of people involved in decision making grows. Jim Azar *et al.* proposed a methodology named value oriented prioritization for prioritizing requirements in small scale firms. Philip *et al*. [16] presented a systematic literature review about the various researches carried out in this field. In all the methodologies, rankings are done either using one criterion or almost two criteria.

ELECTRE has been identified as the best ranking strategy in so many sectors. B. Roy [17] [18] stated the advantage of ELECTRE over different ELECTRE methods and has presented the fact that ELECTRE is best taking decisions with multiple criteria. Juan Carlos, Leyva-Lopez *et al.* discussed about the use of ELECTRE-III in group decision making [9]. Christos Giannoulis, Alessio Ishizaka [8] discussed the advantage of using ELECTRE-III in ranking universities.

## 5. ELECTRE: An Overview

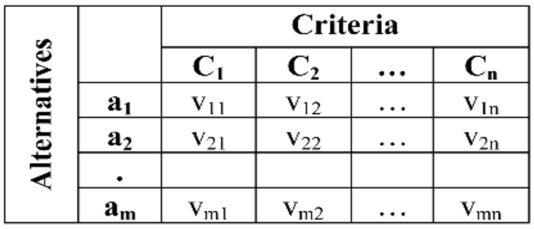
The ELECTRE (Elimination and Choice Translating algorithm) family was introduced by Benayoun, Roy and Sussman in 1968. The method was later developed by Bernard Roy (Roy, 1996). This family includes ELECTRE I, II, III, IV, IS and TRI methods (Vincke, 1992; Roy, 1996). All ELECTRE methods appear to be similar in describing the concepts but differ in type of decision problem being solved. It has been proved that ELECTRE I is found to be best suited for selection problems, ELECTRE TRI seems to be suited for assignment type problems and the other ELECTRE methods are for ranking problems. Especially, ELECTRE III is proved to be more suitable for ranking problematic. ELECTRE III has been identified to be useful in different applications [8] [19] [20].

A typical MCDM (Multiple Criteria Decision Making) problem can be defined as a ranking aid to arrange a finite number of decision alternatives, each of which is clearly described in terms of different characteristics. These characteristics are also often called attributes or decision criteria as in **Figure 1**. A typical MCDM can be represented by using a two dimensional matrix as in **Figure 1**.

Also, for every criterion, every decision maker has to define the following:

1) Preference threshold (p)
2) Indifference threshold (q)
3) Veto thresholds (v) where ($v \geq q \geq p$)
4) Importance rating ($w_j$) for each criterion j.

To start with, ELECTRE requires the evaluation of two indices, the *concordance* index and the *discordance* index, defined for each pair of alternatives. **Figure 2** depicts the main steps of ranking using the ELECTRE III model. In the following subsections, the complete description of the ELECTRE III model has been summarized.



| | | **Criteria** | | | |
|---|---|---|---|---|---|
| | | $C_1$ | $C_2$ | … | $C_n$ |
| **Alternatives** | $a_1$ | $v_{11}$ | $v_{12}$ | … | $v_{1n}$ |
| | $a_2$ | $v_{21}$ | $v_{22}$ | … | $v_{2n}$ |
| | . | | | | |
| | $a_m$ | $v_{m1}$ | $v_{m2}$ | … | $v_{mn}$ |

**Figure 1.** MCDM model, where, $v_{ij}$ represents the value of Alternative *i* with respect to criteria *j*.

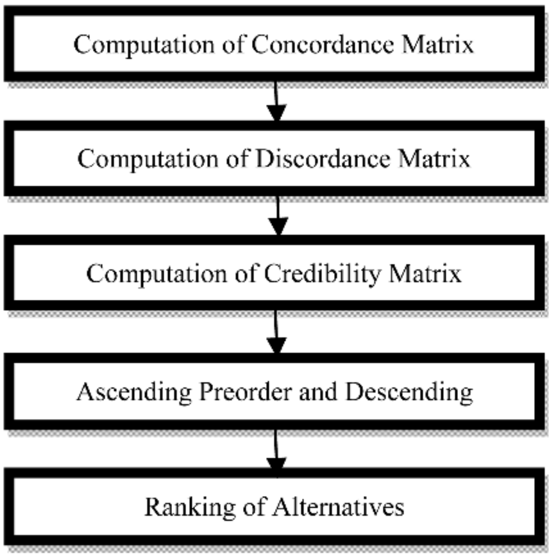

Computation of Concordance Matrix
Computation of Discordance Matrix
Computation of Credibility Matrix
Ascending Preorder and Descending
Ranking of Alternatives

**Figure 2.** ELECTRE steps.

## 5.1. Computation of Concordance Matrix

The strength of the hypothesis that alternative Ai is at least as good as alternative Aj is measured by the concordance index between the pair of alternatives Ai and Aj and is calculated by using Equation (1),

$$C(a,b) = \frac{1}{W}\sum_{j=1}^{n} w_j c_j(a,b) \tag{1}$$

where,

$$W = \sum_{i=1}^{n} w_i \tag{2}$$

$$C_j(a,b) = \begin{cases} 1, & diff \geq -q_j \\ \dfrac{diff + P_j}{P_j - q_j}, & -q_j > diff > -P_j \\ 0, & diff \leq -P_j \end{cases} \tag{3}$$

## 5.2. Computation of Discordance Matrix

The discordance index measures the strength of evidence against the hypothesis and is calculated using Equation (4).

$$D_j(a,b) = \begin{cases} 1, & diff \geq v_j \\ \dfrac{diff - P_j}{v_j - P_j}, & P_j \leq diff \leq q_j \\ 0, & diff \geq P_j \end{cases} \tag{4}$$

## 5.3. Computation of Credibility Matrix

Credibility matrix indicates the reliability of outranking hypothesis. If the concordance index is higher or equal to the discordance index for all criteria, then degree of credibility is equal to concordance index. If the concordance index is strictly below the discordance index, then the degree of credibility is equal to the concordance index lowered in direct relation to the importance of those discordances. Hence,

$$S_j(a,b)=\begin{cases}C(a,b), & \text{if } D_j(a,b)\le C(a,b)\ \forall j\\ C(a,b)\cdot\pi_{i\in j(a,b)}\dfrac{1-D_i(a,b)}{1-C(a,b)}, & \text{otherwise}\end{cases}\tag{5}$$

where $\pi(a,b)$ is the set of criteria for which $D_j(a,b)>C_j(a,b)$.

## 5.4. Ascending Preorder and Descending Preorder

The first pre-order is obtained using descending distillation, by selecting the best rated alternatives initially, and finishing with the worst. The second pre-order is obtained using ascending distillation, selecting the worst rated alternatives initially, and finishing with the best. The two pre-orders which are set based on a qualification score for each alternative as follows:

Step 1: Set $\lambda_0$ equals to the maximum value of $S(a,b)$ in credibility matrix (A) as per the Equation (6).

$$\lambda_0=\max_{a,b\in S}S(a,b)\tag{6}$$

Step 2: A cutoff level of outranking $\lambda_1$ is defined as the largest outranking score which is just less than the maximum outranking score minus the discrimination threshold.

$$\lambda_1=\max_{\{S(a,b)<(\lambda_0-s(\lambda_0))\}\in S}S(a,b)\tag{7}$$

and

$$S(\lambda_0)=\alpha+\beta\lambda\tag{8}$$

where, $s(\lambda_0)$ is the discrimination threshold at the maximum level of outranking $\lambda_0$. The values of $\alpha$ and $\beta$ are usually 0.3 and −0.15 [20].

Step 3: At initial cutoff level, $a$ outranks $b$ if $S(a, b)$ is greater than the cutoff level and $S(a, b)$ exceeds $S(b, a)$ by more than the discrimination threshold satisfying the condition.

Step 4: Every time when $a$ outranks $b$, $a$ is given a score of +1 (strength) and $b$ is given −1 (weakness). For each alternative, the strengths and weaknesses are added together to give a final qualification score.

Step 5: Within Descending Distillation, the alternative with the highest qualification score is assigned to a rank and removed from the procedure and the process is repeated for all remaining options. Within Ascending Distillation, the alternative with the lowest qualification score is assigned to a rank and removed from the procedure and the process is repeated for all remaining options.

The results of the two procedures Descending Distillation and Ascending Distillation are combined to form complete ranking that is consistent with the two procedures.

## 6. Proposed Prioritization Approach

**Figure 3** depicts the web based decision support system built using PHP as frontend and MySQL as backend for performing prioritization. The priorities of requirements are collected from stakeholders using the web interface. Interface has been developed in such a way that the product manager can either send the request either to specified

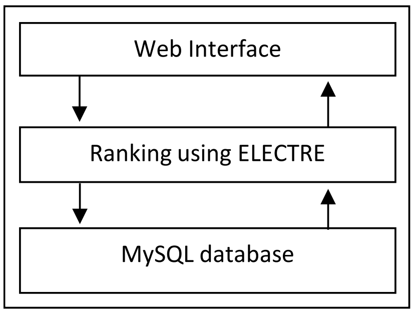


**Figure 3.** Architecture diagram for the proposed system.

group of people in case of Be-spoken product or to a general mass in case of market-driven product. The inputs from the stakeholders are collected using 100-point method [21]. The stakeholders are given the liberty to distribute the 100 points to the requirements according to their perceived priority. The inputs given by stakeholders are stored in MySQL database which is then processed using ELECTRE. The layer where ELECTRE is implemented is independent from the other layers and hence, can be upgradable for similar decision problems.

Our approach for validation of the project starts with selection of a project to assess the effectiveness of the method. After careful analysis, we have decided to take "Library Management System" as the pilot project. The standard set of requirements was identified after having done a detailed literature survey on the topic. In order to ascertain the effectiveness of the proposed method, we have associated ourselves with a web based development company. The project manager of the company was asked to do the project using his own view. In order to standardize it has been planned to have working hours from 9:00 AM to 4:00 PM with a break of 1 ½ hours lunch break in between. Also, after having discussion, it has been decided to use Visual Basic as front end and Ms–Access as backend. The results were recorded that include the number of days taken for completion, number of Lines of Code and customer satisfaction.

As the second step, we have planned to implement the same Library Management System using the proposed method. The set of personnel used to control prioritization was identified which take in 2 developers, 1 project manager, 1 customer and 2 end users one each with and without computer background. Selected persons are asked to prioritize the requirements using 100-point method. The personnel are given the liberty to choose their own criteria for ranking requirements. For e.g., the Project manager may rank the requirements by having cost as his criteria. The end user may rank the requirements by having value as the criteria. Along with the ranking of requirements, the ranking personnel are asked to give indifference, preference and veto thresholds according to their view.

A simple user-friendly model has been developed and the stakeholders are asked to enter the data according to the criteria they have set. The concordance and discordance matrices are calculated and the distillation process is applied. Further, the final ranking is applied and the results are recorded. The comparative results are shown in **Figure 5** and **Figure 7**. The results are discussed in Section 7.

## 6.1. Requirements Set

Through a detailed literature survey about the software, "Library management system", the following requirements are identified.

- R1: Login form for customers.
- R2: Search form for customers.
- R3: Login form for librarian.
- R4: View and Update customer profile.
- R5: Adding customers and library cards.
- R6: Handling Issue and return of media.
- R7: Applying, Receiving, and updating fine details.
- R8: Adding and removing media from library.
- R9: View availability of books.

## 6.2. Generation of Dataset

The dataset is generated by sending the username and password to access the web system to all the possible stakeholders of the system. The stakeholders are record their priorities using 100-point method. In addition to the following threshold values are entered by every stakeholder. Indifference threshold represents the value below which the priorities can be neglected. Preference threshold represents the fact that the priorities above it must be considered for decision making. **Table 1** represents the dataset collected from set of stakeholders.

## 6.3. Credibility Ranking& Distillation

**Table 2** represents the credibility matrix calculated based on concordance and discordance values.

Distillation process is applied using the Equations (6)-(8).

$$\lambda_0 = \max\left(S\left(a,b\right)\right) = 1.0$$

Table 1. Input matrix.

| | | Stakeholders | | | | | |
|---|---|---|---|---|---|---|---|
| | | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** |
| **Requirements** | **R1** | 05 | 20 | 15 | 05 | 10 | 10 |
| | **R2** | 15 | 05 | 04 | 10 | 05 | 05 |
| | **R3** | 10 | 14 | 20 | 15 | 10 | 10 |
| | **R4** | 05 | 15 | 10 | 10 | 10 | 10` |
| | **R5** | 10 | 15 | 18 | 15 | 15 | 20 |
| | **R6** | 10 | 10 | 12 | 10 | 15 | 10 |
| | **R7** | 15 | 07 | 04 | 10 | 10 | 10 |
| | **R8** | 20 | 13 | 12 | 15 | 10 | 20 |
| | **R9** | 10 | 05 | 05 | 10 | 15 | 05 |
| | **Indifference threshold (q)** | 05 | 05 | 05 | 05 | 05 | 05 |
| | **Preference threshold (p)** | 15 | 10 | 10 | 10 | 10 | 10 |
| | **Veto threshold (v)** | 20 | 15 | 15 | 15 | 15 | 15 |
| | **Weight of stakeholder (w)** | 1 | 4 | 4 | 2 | 3 | 3 |

Table 2. Credibility matrix.

| | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** | **R7** | **R8** | **R9** |
|---|---|---|---|---|---|---|---|---|---|
| **R1** | 1.00 | 0.00 | 0.88 | 1.00 | 0.71 | 1.00 | 0.97 | 0.65 | 0.00 |
| **R2** | 0.53 | 1.00 | 0.58 | 0.72 | 0.18 | 0.68 | 1.00 | 0.54 | 0.82 |
| **R3** | 0.95 | 0.00 | 1.00 | 1.00 | 0.82 | 1.00 | 0.00 | 0.79 | 0.00 |
| **R4** | 1.00 | 0.97 | 0.76 | 1.00 | 0.68 | 1.00 | 0.97 | 0.76 | 1.00 |
| **R5** | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.97 | 0.00 |
| **R6** | 0.76 | 1.00 | 0.86 | 1.00 | 0.78 | 1.00 | 1.00 | 0.79 | 1.00 |
| **R7** | 0.53 | 1.00 | 0.67 | 0.81 | 0.45 | 0.86 | 1.00 | 0.64 | 1.00 |
| **R8** | 0.91 | 0.00 | 0.86 | 1.00 | 0.95 | 1.00 | 1.00 | 1.00 | 0.00 |
| **R9** | 0.53 | 1.00 | 0.58 | 0.76 | 0.91 | 1.00 | 0.56 | 0.56 | 1.00 |

$$s(\lambda_0) = 0.3 - 0.15\lambda_0 = 0.15$$

$$\lambda_0 - s(\lambda_0) = 1.0 - 0.15 = 0.85$$

$$\lambda_1 = \max_{\{S(a,b)<(\lambda_0 - s(\lambda_0))\}\in S} S(a,b) = 0.82$$

From the credibility matrix with $\alpha = 0.82$ shown in Table 3, a graph can be drawn by having outgoing edges from each node. Strength of each requirement($r$) is equivalent to the number of outgoing edges from that requirement(r). Similarly, weakness of each requirement is equivalent to the number of incoming edges to that requirement. The qualification is the difference between strength and weakness. Alternatively, the strength can also be found by counting the number of 1's row wise. The weakness can be found by counting number of 1's column wise.

Table 3. Credibility matrix with $\alpha = 0.82$.

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 |
|---|---|---|---|---|---|---|---|---|---|
| **R1** | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| **R2** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **R3** | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| **R4** | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| **R5** | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| **R6** | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **R7** | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| **R8** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| **R9** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

From Figure 4, it is clear that requirement R5 and R8 needs to be implemented first. Since, the process leaves a tie between (R5, R8), (R1, R3, R4, R9) and (R6, R7), a new cut-off value can be found and the process can be repeated.

$$s(\lambda_1) = 0.3 - 0.15\lambda_1 = 0.1725$$

$$\lambda_1 - s(\lambda_1) = 0.85 - 0.1725 = 0.68$$

$$\lambda_2 = \max_{\{S(a,b)<(\lambda_1 - s(\lambda_0))\}\in S} S(a,b) = 0.65$$

From Table 4 and Figure 5, it is inferred that requirement R5 needs to be implemented first followed by R8. Final Ranking is shown in Figure 6.

## 7. Results and Discussion

Requirements engineering, the backbone of software industry, includes various activities like elicitation, analysis, prioritization, etc. Several methods have been proposed by the researchers for prioritization, but only a very few methods support automation along with multi-criteria decision making.

The proposed method is found to be appropriate for the ranking problem due to the following reasons:

- It allows multiple stakeholders to represent their views without the need for face-to-face discussion and hence provides a feel of independency to the stakeholders.
- The comparisons between requirements are not required, hence, make the stakeholders comfortable.
- Thresholds provide a convenient way of ranking requirements.
- It reveals shattering criteria by using veto thresholds.

Figure 7 depicts the time required for completing the project in both methodologies using traditional and proposed approach. The time required for completion when ranking is proper is greatly reduced compared to the implementation time without considering prioritization. This occurs since the views of stakeholders not only include the importance of the requirement, but also the dependencies among the requirement. Projects developed by considering multiple criteria like cost, time, value, dependencies, resource availability ensure proper ordering, reusability and enhanced flow of development process. This greatly reduces the development time and hence improves user satisfaction.

Figure 8 shows the reduction in LOC (Lines of Code) when requirements are properly ranked before implementation starts. The reduction in lines of code is because of the identification of reusable code. The human effort and subsequently the time, cost taken for completion of the project increase. This creates dissatisfaction among customers. The results show that the proposed decision support system using ELECTRE works effectively in solving the problem of prioritization in projects with different categories of stakeholders and that it overcomes the scalability issue.

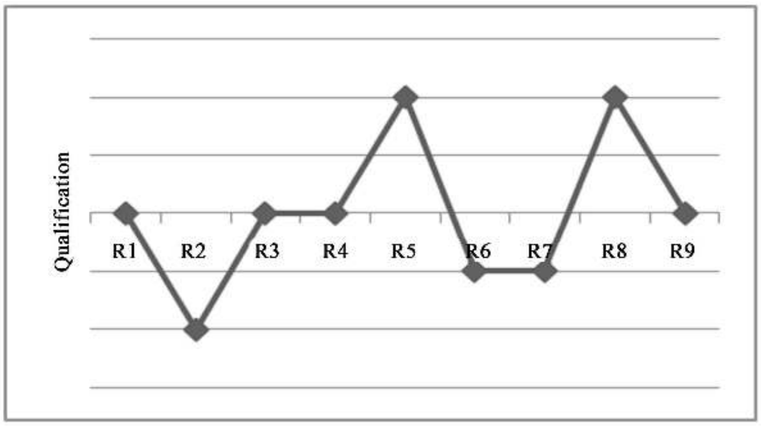


**Figure 4.** Qualification.

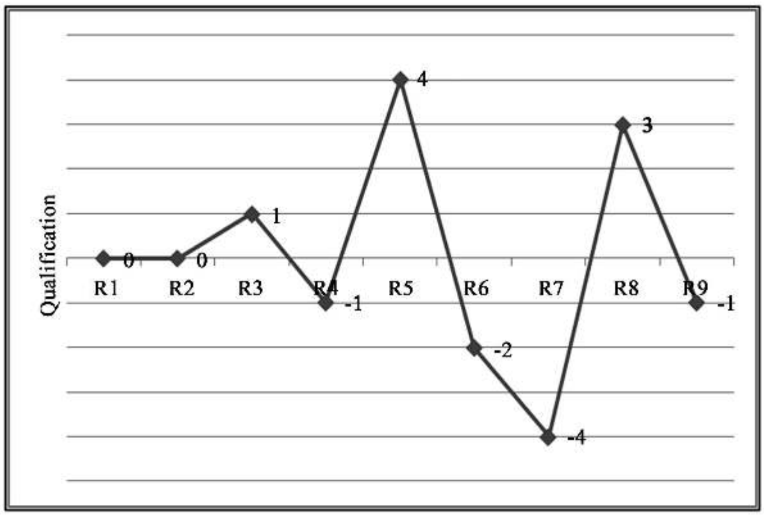


**Figure 5.** Qualification.

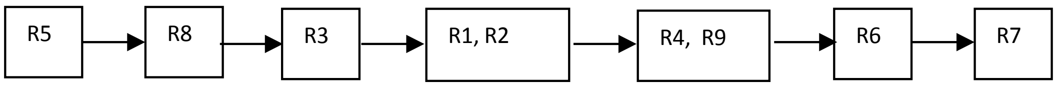


**Figure 6.** Final ranking of requirements.

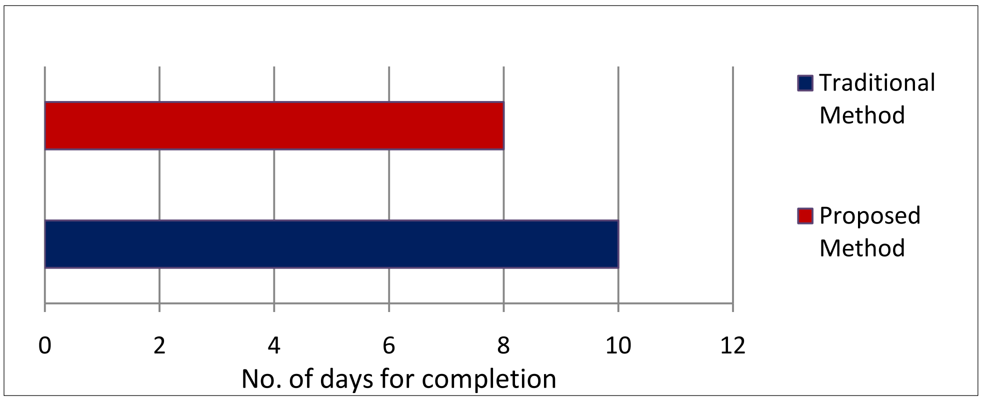


**Figure 7.** Number of days required for completion.

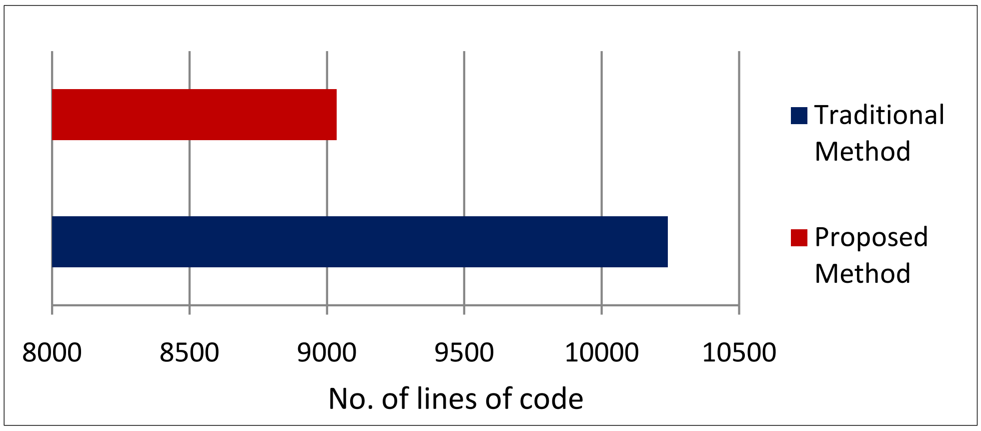


**Figure 8.** Number of lines of code.

**Table 4.** Credibility matrix with $\alpha = 0.65$.

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 |
|---|---|---|---|---|---|---|---|---|---|
| **R1** | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| **R2** | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| **R3** | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| **R4** | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| **R5** | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| **R6** | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| **R7** | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| **R8** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| **R9** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

## 8. Future Works

The method is validated using a case study with fewer requirements. Literature shows that the method is scalable, but the validation is not done in this research work. In the future, the method is planned to be validated with large scale requirements. Exact distribution of 100 points to all the requirements by the stakeholders is not possible in practice. Hence, fuzzy concept is planned to be adopted to collect requirements from stakeholders. In this study, the weightage for stakeholders is given by the product owner. This may create a bias on ranking. In the future, the weightage for the stakeholders will automatically be generated using the results of a generally collected dataset.

## References

[1] Bergman, B. and Klefsjö, B. (2003) Quality from Customer Needs to Customer Satisfaction. Student Literature AB, Lund.

[2] Sillitti, A. and Succi, G. (2005) Requirements Engineering for Agile Methods. In Aurum, A. and Wohlin, C., Eds., *Engineering and Managing Software Requirements*, Springer, Berlin, 315. http://dx.doi.org/10.1007/3-540-28244-0_14

[3] Lehtola, L., Kauppinen, M. and Kujala, S. (2004) Requirements Prioritisation Challenges in Practice. *Proceedings of the 5th International Conference on Product Focused Software Process Improvement*, Kansai Science City, 5-8 April 2004, 497-508. http://dx.doi.org/10.1007/978-3-540-24659-6_36

[4] Karlsson (1995) Towards a Strategy for Software Requirements Selection. PhD Thesis, Linkoping University, Sweden.

[5] Lehtola, L. (2006) Providing Value by Prioritizing Requirements throughout Product Development: State of Practice and Suitability of Prioritization Methods. PhD Thesis, HUT/Departure of Computer Science.

[6] Ruhe, G., Eberlein, A. and Pfahl, D. (2002) Quantitative WinWin—A New Method for Decision Support in Requirements Negotiation. In: *Proceedings of the* 14*th International Conference on Software Engineering and Knowledge Engineering* (*SEKE*'02), ACM Press, New York, 159-166

[7] Thayer, R. and Dorfman, M., Eds. (1997) Software Requirements Engineering. IEEE Computer Society Press, Los Alamitos.

[8] Giannoulis, C. and Ishizaka, A. (2010) A Web-Based Decision Support System with ELECTRE III for a Personalised Ranking of British Universities. *Decision Support Systems*, **48**, 488-497. http://dx.doi.org/10.1016/j.dss.2009.06.008

[9] Leyva-López, J.C. and Fernández-González, E. (2003) A New Method for Group Decision Support Based on ELECTRE III Methodology. *European Journal of Operational Research*, **148**, 14-27. http://dx.doi.org/10.1016/S0377-2217(02)00273-4

[10] Mousseau, V. and Dias, L. (2004) Valued Outranking Relations in ELECTRE Providing Manageable Disaggregation Procedures. *European Journal of Operational Research*, **156**, 467-482. http://dx.doi.org/10.1016/S0377-2217(03)00120-6

[11] Nikula, U., Sajaniemi, J. and Kälviäinen, H. (2000) A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises. Telecom Business Research Center Lappeenranta, Lappeenranta University of Technology, Lappeenranta.

[12] Subash (2006) India's Software Industry. In: Vandana Chandra, V., Ed., *Technology*, *Adaptation and Exports*: *How Some Developing Countries Got It Right*, World Bank, Washington DC, 95-124.

[13] Azar, J., Smith, R.K. and Cordes, D. (2007) Value-Oriented Requirements Prioritization in a Small Development Organization. I*EEE Software*, **2007**, 32-37. http://dx.doi.org/10.1109/MS.2007.30

[14] Wiegers, K.E. (1999) First Thing First: Prioritizing Requirements. *Software Development*, **7**, 48-53.

[15] Racheva, Z., Daneva, M. and Buglione, L. (2008) Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products. *Proceedings of the* 2*nd International Workshop on Software Product Management*, Barcelona, 2008, 49-58.

[16] Achimugu, P., Selamat, A., Ibrahim, R. and Mahrin, M.N. (2014) A Systematic Literature Review of Software Requirements Prioritization Research. *Information and Software Technology*, **56**, 568-585. http://dx.doi.org/10.1016/j.infsof.2014.02.001

[17] Roy, B. (1996) Multicriteria Methodology Goes Decision Aiding. Kluwer Academic Publishers, Berlin. http://dx.doi.org/10.1007/978-1-4757-2500-1

[18] Roy, B. (1990) The Outranking Approach and the Foundations of ELECTRE Methods. In: Bana e Costa, C.A., Ed., *Readings in Multiple Criteria Decision Aid*, Springer-Verlag, 155-183. http://dx.doi.org/10.1007/978-3-642-75935-2_8

[19] Ulubeyli, S. and Kazaz, A. (2009) A Multiple Criteria Decision-Making Approach to the Selection of Concrete Pumps. *Journal of Civil Engineering and Management*, **15**, 369-376. http://dx.doi.org/10.3846/1392-3730.2009.15.369-376

[20] Marzouk, S M.M. (2011) ELECTRE III Model for Value Engineering Applications. *Automation in Construction*, **20**, 596-600. http://dx.doi.org/10.1016/j.autcon.2010.11.026

[21] Leffingwell, D. and Widrig, D. (2000) Managing Software Requirements: A Unified Approach. Addison-Wesley Longman Inc., New York.