International Journal of Software Engineering and Its Application
Vol.12, No.1 (2018), pp.33-46
http://dx.doi.org/10.21742/ijseia.2018.12.1.03

An Integrated Framework for Prioritizing Software Specifications in Requirements Engineering

Ishaya Gambo[1*], Rhoda Ikono[2], Philip Achimugu[3] and Abimbola Soriyan[4]

[1,2,4]*Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria*
[3]*Department of Computer Science, Lead City University, Ibadan, Nigeria*
[1]*ipgambo@gmail.com*, [2]*rhoda_u@yahoo.com*, [3]*check4philo@gmail.com*, [4]*hasoriyan@yahoo.com*

Abstract

Requirements prioritization is an established activity facilitating good decision making for the purpose of coping with complexities that often arise when determining the right choice of requirement during requirements engineering process. However, making precise and accurate decision for the purpose of averting subsequent system failure is an issue of concern when developing large scale systems. In this paper, we proposed the use of imprecise knowledge-based solutions over precise-knowledge based solutions for prioritizing software requirements to overcome the problem of decision making. In this regard, our proposed technique is an integration of Fuzzy Multi Criteria Decision Making (FMCDM), similarity measures and target-based approach to requirements prioritization using linguistic values of triangular fuzzy numbers. With the integrated framework, the challenge of making decisions under uncertain conditions are considered. The integrated framework was based on some defined criteria, a three-phased process comprising of five steps and a five-point scale used to determine the relative values of requirements. The result is the specification of a stepwise process of the computations that can be performed during decision making by the integrated technique proposed. It is hoped that when this technique is implemented, executed, evaluated and validated on case study, a promising results will be achieved. For validating the proposed technique, requirements from real-life case studies can be elicited, analysed, and reconciled for completeness and consistency.

Keywords: *Requirements engineering, prioritization, similarity measures, requirement*

1. Introduction

In life everybody makes one form of decision or the other [1]. In doing this, requirement definition is inevitable. Thus, understanding the complexities inherent in several decision making tasks has become an on-going research effort that requires keen attention for a major breakthrough. Hence, the need to make precise and accurate decision for the purpose of averting subsequent system failure in software development is an issue of concern for the software engineering community. Brooks in his paper remarked that "The hardest single part of building a software system is deciding what to build…. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later." [2]. Brooks observation has become one of the justification for prioritizing software requirements in order to cope with these complex decision making tasks and problem [3]. Requirements prioritization in this context is the process of managing the subjective views and choices of stakeholders as specified in their requirements listing and expectations. This process is one among the several activities of

Received (July 6, 2016), Review Result (November 30, 2016), Accepted (September 12, 2017)

ISSN: 1738-9984 IJSEIA
Copyright ⓒ 2018 SERSC

requirements engineering contributing towards making good decisions for software systems [4], and it is the activity required for the selection of appropriate requirements [5] for implementation. Even in an agile software engineering methodology for engineering sociotechnical systems of systems as observed in [6], requirements prioritisation can be useful in making the right choice from the different viewpoint aspects of stakeholders. Within the premise of goal-oriented requirements engineering methodology [7, 8], prioritisation is essential for the purpose of selecting the goals based on domain specific needs [9]. Requirement engineering (RE) deals with human. The involvement of human makes the processes in RE naturally collaborative in nature. This is as a result of the intensiveness from both knowledge and human perspectives that opens up the problem of decision making [10]. Prioritization in this regard is essential as a first aid approach to overcoming the problem of decision making.

Consequently, most software prioritization techniques are centered on the importance of requirements, and the corresponding value each stakeholder have attributed to the requirement. Therefore, choosing the correct requirements from a list of requirement specifications as elicited from stakeholders' is essential for engendering the development of cost effective quality software. One way of accomplishing this is to prioritize the requirements to enable the selection of optimal set [11].

In this paper, we are proposing the use of imprecise knowledge-based solutions over precise-knowledge based solutions for prioritizing software requirements. This is because, some decision making tasks are quantitatively complex to be understood due to the uncertainties associated with multi-criteria decision making tasks. We found in literature that most Fuzzy Multi Criteria Decision Making (FMCDM) approaches are based on fuzzy set theories. In this regard, our proposed approach is an integration of FMCDM, similarity measures and target-based approach to requirements prioritization using linguistic values of triangular fuzzy numbers.

The paper is further structured as follows: First is an overview of the paper, which is followed by the review of related literatures of some prioritization techniques in Section 2. Next, the methodology describing the approach used and the various components of the integrated proposed approach was presented in Section 3. These components were meant to address the limitations of existing prioritization techniques. In Section 4, the proposed integrated technique was presented alongside the weight scales and computational aspects. The emphasis in this section was on its suitability and relevance for requirements prioritization. Section 5 concludes the paper and suggest some future work for the implementation of the integrated technique.

2. Review of Related Literature

Several benefits of requirements prioritization have been identified in literature to serve as the basis for having the right sets of requirements in a software development project. For instance, Liaskos *et al.*, [12] enumerated some of these benefits and opined that prioritizing software requirement among others, is aimed at handling and negotiating the contradictory and conflicting expectations from each stakeholder. The work of Pitangueira *et al.*, [13] further enumerated the aspects of prioritization most researchers focused on, and also identified the most important prioritization techniques used based on the defined problem(s).

However, there has been other considerable research on the analysis of various issues relating to existing prioritization techniques. For example, Babar et al. [14] observed that prevailing prioritization techniques are deficient in the aspects of handling software projects with large set of requirements specifications. Consequently, this has rendered current techniques unsuitable for prioritizing large scale sets of requirements in a software development project. Another example is the in-depth review that classified existing

 Copyright ⓒ 2018 SERSC

prioritization techniques in [15] and made suggestions on possible areas of improvement based on the disadvantages discovered.

More generally, Dabbagh and Lee [16] opined that the collective results with prioritizing software requirement is an ordering of prioritized lists of requirements that needs to be considered first during the software development process. The authors Achimugu *et al.*, [15], observed that software requirements prioritization is among the design principles that can engender the functionality of any software product consider for development. One of the reasons for this could be tailored towards the possibility of establishing tradeoff among conflicting constraints such as schedule, budget, resources, time to develop, time to market and quality [12].

The work of Pergher and Rossi [17] provided the justification for evaluating prioritization tools by conducting a methodological mapping study. Dabbagh *et al.*, [18] supported this with focus on executing two consecutive controlled experiments that was aimed at evaluating current prioritization techniques. Again, the work of Riņķevičs and Torkar [19] proposed the ECV methods for the analysis of results from commutative voting (CV) technique. The authors focused on the empirical analysis of CV and the corresponding implications.

Furthermore, most existing techniques for prioritizing software requirements accept the association of each requirement with a priority while others group requirements by priority level [3]. Several methods like Cost-Value [20], Quantitative Win-Win [21] and EVOLVE [22] have applied existing prioritization techniques within the scope of a larger project for decision making. Despite the existence and application of these techniques, the complexities during decision making process requires proper understanding due to the uncertainties associated with multi-criteria decision making processes and tasks. In the view of this paper, we make justification that since requirement prioritization is a complex multi-criteria decision making process [23], an integrated approach is required to overcome some of these complexities when deciding on which requirement has the highest priority.

Going further, we noted that the Analytical Hierarchy Process (AHP) has been seen to have the ability of reducing complexities in decision making during requirements prioritization. The AHP approach does this by reducing complex decisions to a series of one-on-one comparisons [24, 25, 26]. The strength of this technique is in its ability to provide adequate support in obtaining the best decision. However, the technique is deficient in terms of scalability when large volume of stakeholders' requirements are subject to decision making process. For example, given n number of requirements, it will require an $n \times (n - 1) / 2$ comparison to be made at each hierarchical level. Obviously, when the number of requirements increases, the number of comparison is equally expected to increase with a magnitude of $O(n^2)$ [27, 28]. Other techniques that suffers the same lack of scalability and difficulties in decision making includes; Round-the-Group Prioritization and Cost-Value Ranking [29, 20], the Planning game technique [30], the Binary-tree [31, 32, 33], Pairwise comparisons and so on. The Attribute goal-oriented requirement analysis technique [34, 35] focused on the computation of preference values assigned to requirements in a decision matrix form represented in a goal graph. Still, this technique is incompetent in the handling of complexities inherent in the goal graph. Therefore, our paper proposes an integrated technique that encapsulate the FMCDM, similarity measures and target based approach in prioritizing software requirements using linguistic values (LV) of triangular fuzzy numbers (TFN).

3. Methodology

The quantitative research approach was considered appropriate in this paper because of its suitability in quantifying the various stakeholders' expectations (or requirements), thereby providing a generalised results from a large number of requirements. This

Copyright ⓒ 2018 SERSC

approach supports fixed design that are highly pre-specified and prepared. It allows the conceptualization of a framework or theory to determine what is been sought for. In this case the fuzzy set theories were conceptualised for proper decision making amidst the various uncertainties. The quantitative design was used to quantify the relationship expected in the integrated approach in order to have results that are collective and prescriptive. With the quantitative approach, the similarities in the LV of TFN assigned to the elicited requirements can be prioritized. This will allow us see the requirements with the highest priority level for implementation within the set time frame of the software development project.

The ranking of requirements can be achieved using fuzzy numbers based on the fuzzy logic concept. A criterion can be computed by summing all the weights and the highest ranked alternatives should be considered first. In most cases, requirements specifications are subjective in nature. Therefore, a fuzzy based approach can be adopted to help in making objective decisions. As specified in the work of Dubois and Prade [36], addressing multiple decisions making is usually achieved using two classifications. The first dealt with the aggregation of weighted scores with respect to each criterion, while the second has to do with ranking the criteria with normalized values.

3.1. Defining Components of the Methodological Approach

Definition 1: Base on the works of Dubois and Prade [37], Zhang [38] and Bárdossy and Duckstein [39], a triangular fuzzy number $\tilde{M}$ can be defined by a triplet denoted as $\tilde{M} = (a,b,c)$ where a ≤ b ≤ c has the following triangular-type membership functions:

$$\mu_M(x) = \begin{cases} \frac{x-a}{b-a}, \; if \;\; a \le x \le b \\ \frac{c-x}{c-b}, \; if \;\; b \le x \le c \\ 0, \quad otherwise \end{cases} \tag{1}$$

From equation 1 above, the maximum value attributed to the fuzzy set $\mu(x)$ is b, while the lower and upper bounds are attributed to a and c respectively. In this context, a and c contains all the linguistic variables in the fuzzy set $\mu(x)$. The triangular fuzzy numbers (TFNs) are depicted in Table 1. The TFNs can be used as alternative scores in order to determine relative values.

Table 1. Triangular Fuzzy Numbers (TFNs)

Variables	Fuzzy Rank	Fuzzy Weights
Extremely Important (EI)	(1,1,1)	(1,1,1)
Weakly More Important (WMI)	(3,1,2)	(1/2, 1, 1/3)
Strongly More Important (SMI)	(3,2,5)	(1/5, 1/2, 1/3)
Very Strongly More Important (VSMI)	(5, 3,7)	(1/7, 1/3, 1/5)
Absolutely More Important (AMI)	(7, 4, 9)	(1/9, 1/4, 1/7)

The defuzzification of TFNs can be achieved using Equation 2. Thus:

$$\overset{\sigma}{W} = \frac{a_{j1}+b_{j2}+c_{j3}}{3} \tag{2}$$

 Copyright ⓒ 2018 SERSC

The membership function $\mu_M(x)$ of TFNs which can be used to describe the level of membership of the elements *M* to the fundamental set *X* is as shown in Figure 1.

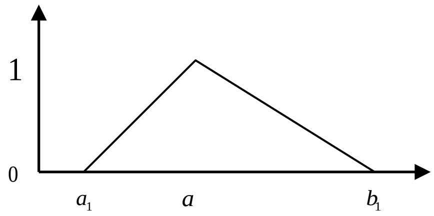

Figure 1. Membership Functions of TFNs

Therefore, any element tending towards 0 connotes that, the member is not included in the given set while the ones tending towards 1 connotes a fully included member. Values strictly between 0 and 1 characterized the fuzzy members.

Definition 2: *Addition, subtraction, multiplication and division of TFNs*

The algebraic operations of triangular fuzzy numbers were executed as follows:

$$(a_1,b_1,c_1) \oplus (a_2,b_2,c_2) = (a_1+a_2, b_1+b_2, c_1+c_2) \tag{3}$$

$$(a_1,b_1,c_1) - (a_2,b_2,c_2) = (a_1-a_2, b_1-b_2, c_1-c_2) \tag{4}$$

$$(a_1,b_1,c_1) \otimes (a_2,b_2,c_2) = (a_1a_2, b_1b_2, c_1c_2) \tag{5}$$

$$(a_1,b_1,c_1) \div (a_2,b_2,c_2) = (a_1 \div a_2, b_1 \div b_2, c_1 \div c_2) \tag{6}$$

Definition 3: *Similarity measures between TFNs*

Let $A=(a_1,a_2,...,a_n)$ and $B=(b_1,b_2,...,b_n)$ be two vectors of n lengths where all the operators are positive. As postulated by Based on Van Laarhoven and Pedrycz [40], the three significant similarity measures are defined as follows:

$$X(A,B) = \frac{CA}{\|C\|_2^2 + \|A\|_2^2 - CA} = \frac{\sum_{i=1}^{n} C_i A_i}{\sum_{i=1}^{n} C_i^2 + \sum_{i=1}^{n} A_i^2 - \sum_{i=1}^{n} C_i \ A_i} \tag{7}$$

$$Y(A,B) = \frac{2CA}{\|C\|_2^2 + \|A\|_2^2} = \frac{2\sum_{i=1}^{n} C_i A_i}{\sum_{i=1}^{n} C_i^2 + \sum_{i=1}^{n} A_i^2 \quad i} \tag{8}$$

$$Z(A,B) = \frac{CA}{\|C\|_2 + \|A\|_2} = \frac{\sum_{i=1}^{n} C_i A_i}{\sqrt{\sum_{i=1}^{n} C_i^2}\ \sqrt{\sum_{i=1}^{n} A_i^2}} \tag{9}$$

The three parameters a_i *(i = 1, 2 and 3)* in TFN $\mu_M(x) = a_1$, a_2 *and* a_3 could be considered as a vector representation with three elements. Based on this notion, the similarity measures between TFNs are shown below:

Copyright ⓒ 2018 SERSC

Let $X=(x_1,x_2,x_3)$ and $Y=(y_1,y_2,y_3)$ be two TFNs where $0 \le x_1 \le x_2 \le x_3 \le 1$ and $0 \le y_1 \le y_2 \le y_3 \le 1$

The similarity measures between the two TFNs are therefore defined as follows:

$$S^A(X,Y)=\frac{\sum_{i=1}^{3} x_i y_i}{\sum_{i=1}^{3} x_i^2+\sum_{i=1}^{3} y_i^2-\sum_{i=1}^{3} x_i\, y_i} \quad (10)$$

$$S^B(X,Y)=\frac{2\sum_{i=1}^{3} x_i y_i}{\sum_{i=1}^{3} x_i^2+\sum_{i=1}^{3} y_i^2} \quad (11)$$

$$S^C=\frac{\sum_{i=1}^{3} x_i y_i}{\sqrt{\sum_{i=1}^{3} x_i^2}\sqrt{\sum_{i=1}^{3} y_i^2}} \quad (12)$$

The similarity measures between the TFNs satisfy the following attributes:

$$0 \le S(X,Y) \le 1; \quad (13)$$

$$S(X,Y)=S(Y,X) \quad (14)$$

If X = Y, that is, $x_i = y_i$, *where i = 1, 2, and 3. S(X, Y) = 1* (15)

Using basic mathematical equation indicated in Equation 16; it can be easily proof that, $S^A(X,Y) \le 1$as shown in Equation 17.

$$2x_i y_i \le x_i^2+y_i^2 \quad (16)$$

$$\sum_{i=1}^{3} x_i^2+\sum_{i=1}^{3} y_i^2-\sum_{i=1}^{3} x_i\, y_i \ge 2x_i y_i - x_i\, y_i \ \ge x_i\, y_i \quad (17)$$

Equations 17 and 10 results in Equations 18 below:

$$S^A(X,Y) \le 1 \quad (18)$$

$$\forall\ X=Y$$

Equation 18 yields 19 as follows:

$$S^A(X,Y)=\frac{\sum_{i=1}^{3} x_i x_i}{\sum_{i=1}^{3} x_i^2+\sum_{i=1}^{3} x_i^2-\sum_{i=1}^{3} x_i\, x_i} \quad (19)$$

***Definition 4**: Target-based model for calculating missing weights of requirements*

Here, the challenge of making decisions under uncertain conditions was considered. In the case where the weights of requirements are imprecisely or not even given by the

 Copyright ⓒ 2018 SERSC

stakeholders, the results can lead to the generation of inhomogeneous decision matrix. These weights could be crisp numbers, an interval value or a fuzzy quantity. To address this challenge, a fuzzy set based technique with the help of extension principles, which culminates into target-based procedure can be employed. To initiate the target-based procedure for solving requirements with vague or missing weights, the inhomogeneous decision matrix must be transformed into the probabilities of meeting the target using Equation 20.

$$P = \sqrt{\frac{1}{R}\sum_{j=1}^{r} W_j} \tag{20}$$

4. The Proposed Integrated Technique

The proposed integrated technique for prioritizing software requirements will be based on the following:

1. Identification of stakeholders' preferences on what the system should do, providing a ranking of the requirements, and taking into consideration the stakeholders' subjective views and importance;
2. Obtain a ranking of requirements that are aimed at determining the preferential requirements of stakeholders;
3. Obtain a cumulative or global rating of these requirements across project stakeholders in order to plan for software release phases.
4. Allow software engineers to perform what-if analysis regarding changes in priorities for the implementation of requirements, while considering new candidate alternatives (*e.g.,* addition or subtraction of requirements).

The first point is to articulate the stakeholders' goals, which will help to solve the potential differences and disagreements in opinions that can arise in terms of the expected requirement specification of the to-be system. For instance, security and response time in most cases are usually the conflicting quality attributes requiring attention. For instance, if the expectation is for a strongly secured system, then it may well be that the response time will increase. The qualities that conflict each other are fundamental information in a trade-off analysis. In such cases, we should look at the initial stakeholders' desires, and check which of the conflicting qualities is ranked higher. This ranking can, from thereafter, be used as a guide by the developers to try their best to satisfy the stakeholders' wishes. This, together with the other potential conflicting qualities, should guide the information to be passed over to the architect.

The second point is about having available trustworthy information from where architects and developers can stem a first vision of the software. System qualities are the criteria with higher impact on the architecture. Therefore, it is necessary to identify and prioritize those we need to address. The main difficulty the architects face is to choose the requirements, or combination of requirements, that best satisfies the set of chosen qualities. The choice of the wrong requirements, or combination of requirements, can bring serious problems throughout the software life-cycle, having a strong impact on the success or failure of the system. Different requirements may pull the system in various directions; each quality leads to a number of implementation strategies, each one satisfying the system requirements and serving the stakeholder needs with varying levels of stakeholder satisfaction.

The third point addressed questions that are related with the identification and display of prioritised requirements across stakeholders. This can be achieved by transforming local weights into global weights. The ranking for potential candidate requirements to be

Copyright ⓒ 2018 SERSC

developed may support stakeholders in identifying new software attributes with less risks and costs.

The fourth point is tailored towards giving developers the flexibility to play with different choices and understand their impact on the ranking and, consequently, on the decisions they have to make. This was very important, because, it will enable the storing of decisions and their respective rationale, which might be useful in future similar decision processes.

Therefore, in Figure 2, we describe the conceptual framework of the proposed integrated technique. This technique is a three-phased process that is composed of five steps. The first step is meant for the elicitation of requirements.

In the second step, trade-offs between criteria using pairwise comparisons can be performed. In addition, the second step supports the calculation of criteria priority vector, requirements with missing weights using the harmonic mean. The normalization of the respective weights and calculation of the weighted normalized matrices can also be carried out in the second step. In step 3, support for the calculation of relative closeness degrees of requirements and consistency indexes is provided. In step 4, the calculation of the weights and the display of prioritized requirements with respect to each criterion, using the classical weighted normalized decision matrix can be dealt with. It is in this step that the ratings for each requirement is/are provided. Finally, step 5 will deal with the performance evaluation of proposed integrated technique.

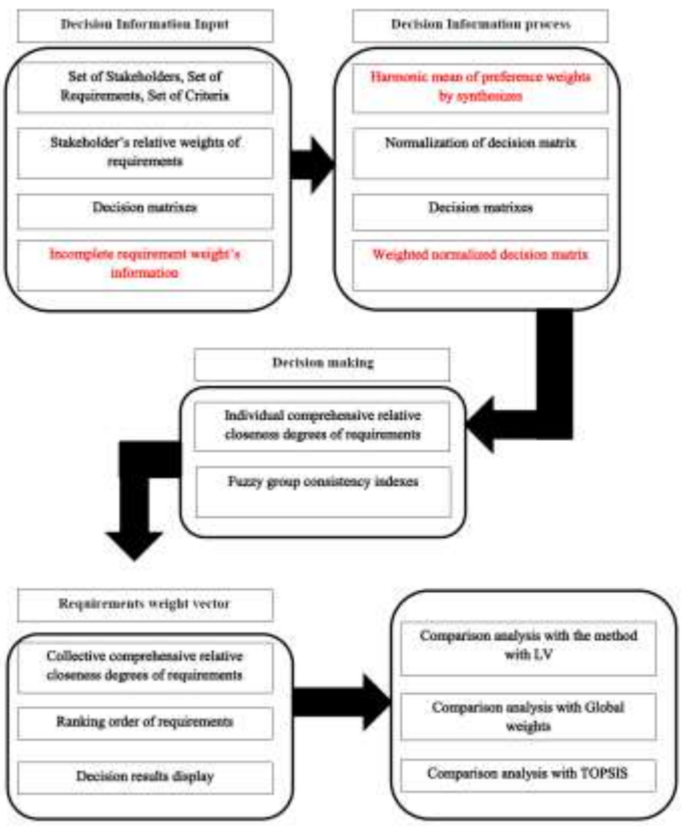

Figure 2. Proposed Integrated Technique

4.1. The Weight Scale

A 5-point scale can be utilised to determine the relative values of requirements. The scale consists of linguistic values and their corresponding scores were used to rate a

Copyright ⓒ 2018 SERSC

typical multi-criteria decision making problem, where the criteria are inversions and indiscrimination. Inversion occurs when the real rankings of requirements are known and the method being tested ranks them in a different order. For example, if the real ranking of three requirements is equal to (5>3>1) then the possible tested method result can be (5>1>3). Indiscrimination happens when the tested method yields a tie between two or more requirements. For example, if the computed result is (1>3=2).

In literature, Triantaphyllou tested 78 different scales. These were classified as Class 1, Class 2…, Class 78, and they all depicts different performances in terms of the two above mentioned criteria, and also on the number of criteria used. However, the 5-point scale suggested in this paper is with a corresponding triangular fuzzy numbers. This is required for use in fuzzy or uncertain environment, which covers the majority of the sizes of sets of criteria that considers equal weights for the criteria shown in Table 2.

Table 2. Weight Scale

Terms	RALCI rating	Equivalent TFNs
Extremely high (EH)	5	(0.9, 1.0,1.0)
Very high (VH)	4	(0.7, 0.9, 1.0)
High (H)	3	(0.5, 0.7, 0.9)
Fair (F)	2	(0.3, 0.5, 0.7)
Low (L)	1	(0.1, 0.3, 0.5)

4.2. Computational Aspects of Proposed Technique

This section specifies the stepwise process of the computations that can be performed by the integrated technique proposed. The first step of phase 1 is meant to identify the criteria and the requirements. The theoretic employed included letting n_c be the number of criteria and n_r the number of requirements. In the second step, stakeholders' weights can be elicited for the relative importance of each requirement based on their respective criteria.

The values of the weight scales are the elements of the criteria matrix c of dimension n_c. For all i and j with $1 \leq i < j \leq n_c$; the i^{th} and j^{th} criteria are compared, leading to the value c_{ij}. The remaining entries of the matrix are determined by $c_{ji}=1/c_{ij}$, *i.e.*, the inverse function.

In the third step, the model calculates the priority vector for the requirements, which represents the weights for phase 2, using the geometric mean. The geometric mean is selected, following the conclusions of Dong *et al.*, [41] about its superior suitability over Saaty's eigenvalue method.

Assuming that s comparisons were done for each attribute, where $s < n$ (preferably $s << n$). These s attributes are drawn randomly and evenly distributed such that each attribute is compared pairwise with some other attribute $2s$ times. The result of stakeholders' assessments is then an $n \times n$ matrix A where $2sn$ of the entries are filled in. In other words, assuming the attribute i is directly compared with the attributes $j_1, \ldots, j_{2s}$ (where the attributes $j_1, \ldots, j_s$ are assigned to i and $j_{s+1}, \ldots, j_{2s}$ are the ones to which i is assigned). The entry a_{ji} in A is the score i achieved when compared with j and $a_{ij} = -a_{ji}$. The entries in row i in A then are all a_{ij} for $j = j_1, \ldots, j_{2s}$, denoted by $a_{i(j1)}, \ldots, a_{i(j2s)}$. The remaining entries are not filled in, except $a_{ii} = 0$.

Progressively, some consecutive repeated calculations can be performed to find a final weight vector. The "first" weight vector is given by the arithmetic mean of these scores. That is, the attribute i gets the following weight;

$$w_1(i) = w(i) = (a_{i(j1)} + \cdots + a_{i(j2s)})/2s$$

For $i = 1, \ldots, n$, and the weight vector is $W_1 = (w_1(1) \ldots w_1(n))^T$.

This is then repeated with the weights of the directly connected attributes taken into account, that is, $w_1(j_l), l = 1, \ldots, 2s$ is added to the value $a_{i(jl)}$. The score attained for each of the directly connected attributes will enhance the relative score and the "second" weight for *i* is

$$w_2(i) = (a_{i(j1)} + w_1(j_1) + \cdots + a_{i(j2s)} + w_1(j_{2s}))/2s) = w_1(i) + (w_2(j_1) + \cdots + w_2(j_{2s}))/2s.$$

This can be done repeatedly until the weight vector stabilizes. That is, W_r does not differ significantly from W_{r+1} and the ranking does not change. Assuming this happens after *r* iterations. Then the attribute *i* has the score/weight.

$w_r(i) = w_1(i) + (w_{r-1}(j_1) + \cdots + w_{r-1}(j_{2s}))/2s$, and the weight vector is $W_r = (w_r(1), \ldots, w_r(n))^T$. In actual sense, the weight vector was expected to stabilize after a relatively small number of iterations.

In this regard, an attribute will only relate to its closest neighbours, but will be pushed up or down on the ranking according to the score attained by the direct neighbours. For example, one attribute, which is expected to score average compared with the others, do really well in the first round and end up towards the top of the ranking if its direct neighbours typically obtain low ratings. The score for such attribute can then be adjusted in the subsequent rounds according to how its neighbours score. Hence, these adjustments will spread out and draw scores from all the attributes and by repeating a satisfactory number of times given by the desired result. In this example the score for this particular attribute can be pulled down.

4.3. Architecture of the Proposed Integrated Technique

In Figure 3, the flowchart depicting the architecture of the proposed technique is depicted. The architecture summarises the proposed steps of prioritizing software requirements based on the definitions and operations indicated. The summary is as enumerated below:

Step 1: Elicit requirements from project stakeholders and determine the criteria for describing each requirement to enable a comprehensive rating exercise.

Step 2: Obtain the decision matrix D as shown in Equation 17.

$$\tilde{A}^k = \left[\tilde{a}_{ij}\right]^k \tag{21}$$

$\tilde{A}^k$, represent the computed decision matrix of the form a_{ij}, b_{ij}, c_{ij} ; *k* stands for the numbers of relevant stakeholders while $\tilde{a}_{ij}$ is the fuzzified local weights of the entire requirements allotted by relevant stakeholders.

Step 3: Determining the relative weights of requirements across project stakeholders using the linguistic scale shown in Table 2. The processes involved in achieving this are described below:

- Construction of a decision matrix by taking the criteria of requirements into consideration.
- Using the stakeholders' weights to fill the decision matrix.
- Finding the weights of each requirement and computing the eigenvalue of the decision matrix.

Step 4: The linguistic values are converted to the corresponding triangular fuzzy numbers across the project stakeholders as represented below:

$$V^k = \left[\langle R_1(a_{ij})\rangle\right]^k \tag{22}$$

 Copyright ⓒ 2018 SERSC

$$V^{k} = \left[\langle R_2 (a_{ij})\rangle\right]^{k} \tag{23}$$

$$V^{n} = \left[\langle R_n (a_{ij})\rangle\right]^{m} \tag{24}$$

Step 5: The weight vector of the stakeholders $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ for each requirement R_i is used to calculate the preference vectors as follows:

$$V_i = \left[\langle R_1 \sum_{k=1}^{m} \alpha_k x_{i\ 11}^{k}, \sum_{k=1}^{m} \alpha_k x_{i\ 12}^{k} \sum_{k=1}^{m} \alpha_k x_{i\ 13}^{k}\rangle\right]^{k} \tag{25}$$

$$\left[\langle R_2 \sum_{k=1}^{m} \alpha_k x_{i\ 21}^{k}, \sum_{k=1}^{m} \alpha_k x_{i\ 22}^{k} \sum_{k=1}^{m} \alpha_k x_{i\ 23}^{k}\rangle\right]^{k} \tag{26}$$

$$\left[\langle R_n \sum_{k=1}^{m} \alpha_k x_{i\ n1}^{k}, \sum_{k=1}^{m} \alpha_k x_{i\ n2}^{k} \sum_{k=1}^{m} \alpha_k x_{i\ n3}^{k}\rangle\right]^{k} \tag{27}$$

Step 6: Normalization of the decision matrix weights is executed by applying Equation 28 and 29.

$$\overline{w}_j = \frac{w}{\sum_{j=1}^{n} w_j} \quad i = 1,..n;\ j = 1,.m \tag{28}$$

$$\overline{w}_j = \begin{bmatrix} \overline{x}_{11}\ \overline{x}_{12} \ldots \overline{x}_{1M} \\ \overline{x}_{21}\ \overline{x}_{22} \ldots \overline{x}_{2M} \\ \vdots \quad \vdots \quad \vdots \\ \overline{x}_{N1}\ \overline{x}_{N2} \ldots \overline{x}_{NM} \end{bmatrix} \tag{29}$$

Where $\sum_{j=1}^{n} \overline{w}_j = 1$

Step 7: Calculation of the weighted normalized decision matrix (WNDM) using Equation 30 and 31.

$$\hat{x}_{ij} = \overline{x}_{ij} * w_j \quad i = 1,..n;\ j = 1,.m \tag{30}$$

$$\hat{w}_j = \begin{bmatrix} \hat{x}_{11}\ \hat{x}_{12} \ldots \hat{x}_{1M} \\ \hat{x}_{21}\ \hat{x}_{22} \ldots \hat{x}_{2M} \\ \vdots \quad \vdots \quad \vdots \\ \hat{x}_{N1}\ \hat{x}_{N2} \ldots \hat{x}_{NM} \end{bmatrix} \tag{31}$$

Step 8: After calculating the WNDM, the next step is to synthesize the weights using Equation 32.

$$\varpi = \frac{\hat{x}_{j1} + \hat{x}_{j2} + \hat{x}_{j3}}{3} \tag{32}$$

Step 9: Requirements are finally prioritized based on the final ranks calculated with the help of Equation 33.

$$P = \sum_{j=1}^{k} \sqrt{\frac{\varpi_{ij}}{k}} \tag{33}$$

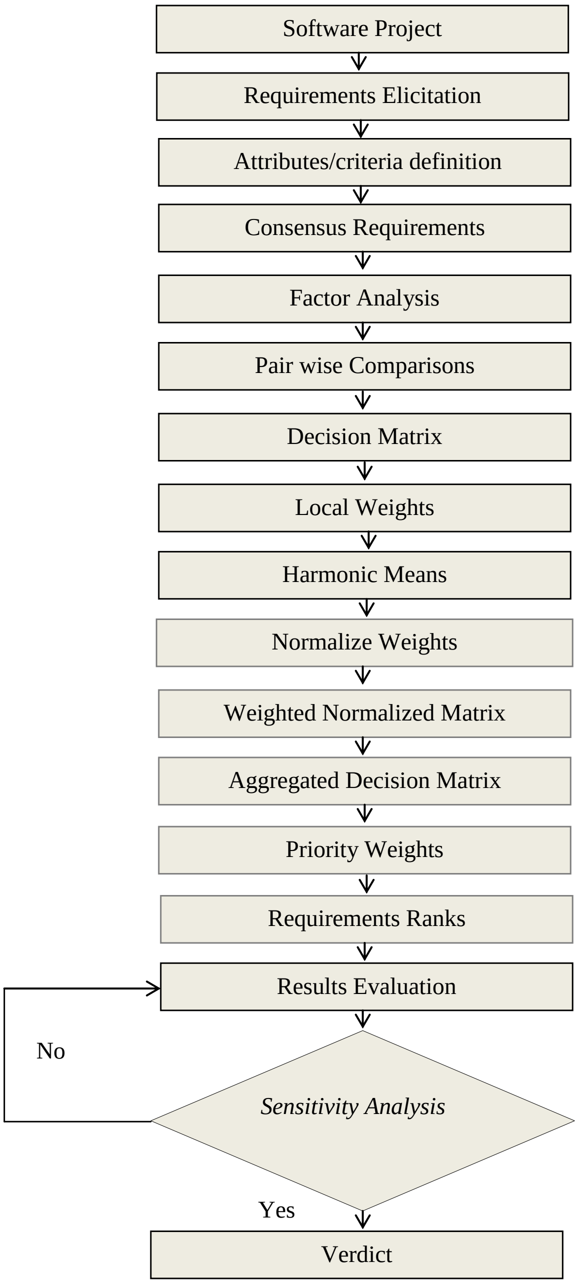

Figure 3. Flowchart of the Proposed Technique

5. Conclusion and Future Work

In conclusion, this paper proposed an integrated technique comprising of Fuzzy Multi Criteria Decision Making (FMCDM), similarity measures and target-based approach to requirements prioritization using linguistic values of triangular fuzzy numbers. The paper

 Copyright ⓒ 2018 SERSC

enumerated the components of the proposed technique. These components were meant to address the limitations of existing prioritization techniques. Consequently, the components of the proposed technique have the capacity of prioritizing large numbers of requirements, reduce disparities or disagreement between ranked weights, reverse ranks when weights or requirements evolves, reduce computational complexities, easy to use and generation of accurate results. Various algorithms and models were formulated in order to enhance the usability of the proposed technique. The evaluation of the proposed technique can be executed with relevant datasets for real-life software development project and validation in the appropriate domain. Therefore, for future research, we look forward to implementing this concept encapsulated in the integrated technique proposed for requirements prioritization. It is hoped that when this technique is implemented, executed, evaluated and validated, a promising results will be achieved.

References

[1] I. P. Gambo, "Development of a Model for Conflict Resolution in the Requirements Engineering Process of Software Systems", Ph.D. thesis, Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria, **(2016)**, pp. 1-298.

[2] F. P. Brooks, "The Mythical Man-Month", Essays on Software Engineering, Addison-Wesley Longman, Boston, MA, USA, **(1995)**.

[3] P. Berander and A. Andrews, "Requirements prioritization", Engineering and managing software requirements, Springer Berlin Heidelberg, **(2005)**, pp. 69-94.

[4] Q. Ma, "The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review", Doctoral dissertation, Auckland University of Technology, **(2009)**.

[5] R. K. Chopra, V. Gupta and D. S. Chauhan, "Experimentation on Accuracy of Non Functional Requirement Prioritization Approaches for different complexity Projects", Perspectives in Science, **(2016)**, http://dx.doi.org/10.1016/j.pisc.2016.04.001.

[6] K. Taveter and A. Norta, "Agile Software Engineering Methodology for Information Systems' Integration Projects", In Proceedings of the 4th International Conference on Future Data and Security Engineering (FDSE 2017), Ho Chi Minh City, Vietnam, Springer, **(2017)** November 29-December 1, pp. 215-230.

[7] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja and P. Giorgini, "Goal-oriented requirements engineering: an extended systematic mapping study", In Requirements Engineering Journal, Springer London, **(2017)**, pp. 1-28.

[8] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja and P. Giorgini, "Goal-oriented requirements engineering: a systematic literature map", In Proceedings of the 24th IEEE International Conference on Requirements Engineering (RE), Beijing, China, IEEE, **(2016)** September 12-16, pp. 106-115.

[9] N. U. Rehman, S. Bibi, S. Asghar and S. Fong, "Comparative Study of Goal-Oriented Requirements Engineering", In Proceedings of the 4th IEEE International Conference on New Trends in Information Science and Service Science (NISS), TBD Gyeongju, Korea, **(2010)** May 11-13, pp. 248-253.

[10] I. P. Gambo, H. A. Soriyan and R. N. Ikono, "A Proposed Process Model for Requirements Engineering using Delphi Techniques for Prioritisation", International Journal of Information Technology and Computer Science (IJITCS), vol. 7, no. 1, **(2014)**, pp. 73.

[11] P. Berander, "Prioritisation of Stakeholder Needs in Software Engineering, Understanding and Evaluation", Blekinge Institute of Technology. ISBN: 91-7295-052-8. Licentiate Series No 2004:12, **(2004)**.

[12] S. Liaskos, S. A. McIlraith, S. Sohrabi and J. Mylopoulos, "Representing and reasoning about preferences in requirements engineering", Requirements Engineering, vol. 16, no. 3, **(2011)**, pp. 227-249.

[13] A. M. Pitangueira, R. S. P. Maciel and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature", Journal of Systems and Software, vol. 103, **(2015)**, pp. 267-280.

[14] M. I. Babar, M. Ghazali, D. N. Jawawi, S. M. Shamsuddin and N. Ibrahim, "PHandler: An expert system for a scalable software requirements prioritization process", Knowledge-Based Systems, vol. 84, **(2015)**, pp. 179-202.

[15] P. Achimugu, A. Selamat, R. Ibrahim and M. N. R. Mahrin, "A systematic literature review of software requirements prioritization research", Information and Software Technology, vol. 56, no. 6, **(2014)**, pp. 568-585.

[16] M. Dabbagh and S. P. Lee, "An approach for prioritizing NFRs according to their relationship with FRs", Lecture Notes on Software Engineering, vol. 3, no. 1, **(2015)**, pp. 1-5.

[17] M. Pergher and B. Rossi, "Requirements prioritization in software engineering: a systematic mapping study", Paper presented at the 2013 IEEE Third International Workshop on Empirical Requirements Engineering (EmpiRE), **(2013)**, pp. 40-44.
[18] M. Dabbagh, S. P. Lee and R. M. Parizi, "Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches", Soft Computing, **(2015)**, pp. 1-24.
[19] K. Riņķevičs and R. Torkar, "Equality in cumulative voting: A systematic review with an improvement proposal", Information and Software Technology, vol. 55, no. 2, **(2013)**, pp. 267-287.
[20] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements", IEEE Software, vol. 14, no. 5, **(1997)**, pp. 67-74.
[21] G. Ruhe, A. Eberlein and D. Pfahl, "Quantitative WinWin: a new method for decision support in requirements negotiation", Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02), ACM Press, New York, **(2002)** July, pp. 159-166.
[22] D. Greer and G. Ruhe, "Software release planning: an evolutionary and iterative approach", Information and Software Technology, vol. 46, no. 4, **(2004)**, pp. 243-253.
[23] D. Leffingwell and D. Widrig, "Managing Software Requirements – A Unified Approach", Addison-Wesley, Upper Saddle River, NJ, USA, **(2000)**.
[24] P. F. Rad and G. Levin, "Project portfolio management tools and techniques", www. iil. com/publishing, **(2006)**.
[25] S. Koul and R. Verma, "Dynamic Vendor selection: a fuzzy AHP approach", International Journal of Manufacturing Technology Management, vol. 22, no. 8, **(2009)**, pp. 1-13.
[26] R. Verma and S. Koul, "Dynamic Vendor Selection: A Fuzzy AHP Approach", International Journal of the Analytic Hierarchy Process, vol. 4, no. 2, **(2012)**. DOI: http://dx.doi.org/10.13033/ijahp.v4i2.25.
[27] J. R. Hubbard, "Theory and Problems of Data Structures with C++", McGraw-Hill, NY, USA, **(2000)**.
[28] N. W. Kassel and B. A. Malloy, "An approach to automate requirements elicitation and specification", Proceedings of 7th International Conference on Software Engineering and Applications, 2003, Marina del Rey, USA, **(2003)** November 3-5, pp. 3-5.
[29] S. Hatton, "Choosing the right prioritisation method", Proceedings of ASWEC 19th Australian Conference on Software Engineering, Australia, **(2008)** March 26-28, pp. 517-526.
[30] C. Duan, P. Laurent, J. Cleland-Huang and C. Kwiatkowski, "Towards automated requirements prioritization and triage", Requirements engineering, vol. 14, no. 2, **(2009)**, pp. 73-89.
[31] M. Aasem, M. Ramzan and A. Jaffar, "Analysis and Optimization of Software Requirements Prioritisation Techniques", Proceedings of IEEE International Conference on Information and Emerging Technologies (ICIET), Karachi, Pakistan, **(2010)** June 14-16, pp. 1-6.
[32] M. Beg, R. Verma and A. Joshi, "Reduction in number of comparisons for requirement prioritisation using B-Tree", Proceedings of IEEE International Advance Computing Conference, (IACC, 2009), Delhi, India, **(2009)** March 6-7, pp. 340-344.
[33] B. Babar, M. Ramzan and S. Ghayyur, "Challenges and Future Trends in Software Requirements Prioritisation", Proceedings of the 2011 IEEE International Conference on Computer Networks and Information Technology (ICCNIT), Abbottabad, Pakistan, **(2011)** July 11-13, pp. 319-324.
[34] H. Kaiya, H. Horai and M. Saeki, "AGORA: Attributed goal-oriented requirements analysis method", Proceedings of the IEEE Joint International Conference on Requirements Engineering, September 9-13, **(2002)**, pp. 13-22.
[35] N. Carod and A. Cechich, "Requirements Prioritization Techniques", Encyclopedia of Information Science and Technology, **(2009)**, pp. 3283-3291.
[36] D. Dubois and H. Prade, "New results about properties and semantics of fuzzy set-theoretic operators", In Fuzzy Sets, Springer US, **(1980)**, pp. 59-75.
[37] D. Dubois and H. Prade, "Operations on fuzzy numbers", International Journal of Systems Science, vol. 9, no. 6, **(1982)**, pp. 613-626.
[38] G. Q. Zhang, "Convergence of a sequence of fuzzy number-valued fuzzy measurable functions on the fuzzy number-valued fuzzy measure space", Fuzzy Sets and Systems, vol. 57, no. 1, **(1993)**, pp. 75-84.
[39] A. Bárdossy and L. Duckstein, "Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems, Boca Raton: CRC Press.Barki. H., and Hartwick,. (2001). Interpersonal conflict and its management in information system development", MIS Quarterly, vol. 25, no. 2, **(1995)**, pp. 195-228.
[40] P. J. M. Van Laarhoven and W. Pedrycz. "A fuzzy extension of Saaty's priority theory". Fuzzy sets and Systems, vol. 11, no. 1, **(1983)**, pp. 199-227.
[41] Y. Dong, Y. Xu, H. Li and M. Dai, "A comparative study of the numerical scales and the prioritization methods in AHP", European Journal of Operational Research, vol. 186, **(2008)**, pp. 229-242.

Copyright ⓒ 2018 SERSC