# A Preference Weights Model for Prioritizing Software Requirements

Philip Achimugu, Ali Selamat*, and Roliana Ibrahim

UTM-IRDA Digital Media Centre,
K-Economy Research Alliance & Faculty of Computing,
Universiti Teknologi Malaysia, Johor Bahru, 81310, Johor, Malaysia
check4philo@gmail.com, {aselamat,roliana}@utm.my

**Abstract.** Software requirements prioritization is the act of ranking user's requirements in order to plan for release phases. The essence of prioritizing requirements is to avoid breach of contract, trust or agreement during software development process. This is crucial because, not all the specified requirements could be implemented in a single release due to inadequate skilled programmers, time, budget, and schedule constraints. Major limitations of existing prioritization techniques are rank reversals, scalability, ease of use, computational complexities and accuracy among others. Consequently, an innovative model that is capable of addressing these problems is presented. To achieve our aim, synthesized weights are computed for criteria that make up requirements and functions were defined to display prioritized requirements based on the global weights of attributes across project stakeholders. An empirical case scenario is described to illustrate the adaptability processes of the proposed approach.

**Keywords:** Software, requirements, weights, prioritization, model.

## 1 Introduction

Prioritization is the act of determining an ordered set of requirements based on their perceived importance by project stakeholders so as to plan for software release phases [1-3]. It is considered to be a multi-criteria decision making process. Essentially, the basic components of decision-making problems are: goal/objective goal, criteria/factors or alternatives/actions. During decision making, many unambiguous criteria are used to elect best alternatives from a pool. These criteria could either be quantitative, qualitative or both. The ranking of alternatives are finally achieved through weighting scales which contain values that decision makers use to determine the relative importance of alternatives.

Decision making in software development process is inevitable if the proposed software must satisfy or meet user's requirements and delivered within time and budget. This is a crucial aspect of software development because; clients specify too many

* Corresponding author.

D. Hwang et al. (Eds.): ICCCI 2014, LNAI 8733, pp. 30–39, 2014.

requirements for implementation without considering the availability of time and resource constraints. Therefore, a meticulously selected set of requirements must be considered for implementation with respect to available resources [4]. The process of selecting preferential requirements for implementation is the most prominent attributes of requirements prioritization techniques. This process aims at determining the most valued or prime requirements from a set or pool of specified requirements [5].

The main aim of this study is to develop a preference based multi-criteria decision making (MCDM) model, capable of enhancing stakeholder's quest for prioritizing requirements with respect to multiple criteria and diverse criteria priorities. A rank is computed by collating all the weights for the requirements in a set with respect to the total number of stakeholders involved in the ranking process. Software products that are developed based on prioritized requirements can be expected to have a lower probability of being rejected.

To prioritize requirements, stakeholders will have to relatively compare them in order to determine their relative value through preference weights [6]. These comparisons grow with increase in the number of requirements [7]. State-of-the-art prioritization techniques such as AHP and CBRanks seem to demonstrate high capabilities [8]. These techniques have performed well in terms of ease of use and accuracy but, still lacking in various areas. In this paper, an enhanced approach for software requirements prioritization is proposed based.

The rest of the article is structured as follows: Section 2 discusses the related work while section 3 enumerates the proposed approach. Section 4 presents an empirical example in order to evaluate the performance of the proposed approach. Section 5 deals with the experimental result which lead to conclusion and future work in section 6.

## 2 Related Work

In literature, many techniques have been proposed for prioritizing software requirements during development processes but most of these techniques are not easily adoptable due to one limitation or the other. Analytic hierarchy process (AHP) seems to be the most widely adopted technique for prioritizing alternatives including software requirements but this prominent technique suffer serious scalability challenges. It cannot prioritize large number of requirements and it is said to be time consuming [9-11]. However, attempts have been made to address the scalability challenges inherent in AHP. For example techniques like binary priority list [12], Case based ranking [13], EVOLVE [14], fuzzy based requirements prioritization approaches [15, 16], Pair wise analysis [17], TOPSIS [18, 19] and fuzzy based MCDM approaches [20, 21] among others. Again, none of these techniques is flawless. Techniques like binary priority list and fuzzy AHP did not cater for dependencies that could exist among requirements before prioritizing them. Case based rank is limited in its inability to support coordination among different stakeholders through negotiations, EVOLVE is reported to be computationally complex along with pair wise comparisons which is also known for producing unreliable results. TOPSIS on the other hand do not possess the ability of updating rank status whenever requirements evolves while the fuzzy based approaches suffer

generally from three major setbacks which include: requirements dependency issues as well as lack of implemented tool to support the proposed approaches. Also, validations of these techniques with real-world projects have not been achieved yet. A detailed analysis and descriptions of existing prioritization techniques with their limitations can be found in [22]. Nonetheless, obvious limitations that cut across existing techniques ranges from rank reversals to scalability, inaccurate rank results, increased computational complexities and unavailability of efficient support tools.

## 3 The Proposed Approach

The main objective of this study is to propose an approach that supports stakeholders in ranking software requirements. In this context, the proposed steps are described as follows:

Step 1: Decompose the problem into a hierarchy of requirements and their interrelated attributes;
Step 2: Generate input data consisting of preference weights of attributes across all stakeholders;
Step 3: Synthesize each subjective judgments and compute the global weights;
Step 4: Calculate the final weights to display prioritized requirements.

The proposed prioritization approach consists of defining a common hierarchy of requirements with their respective attributes to aid comparison by all the stakeholders involved in the software development project (Figure 1).

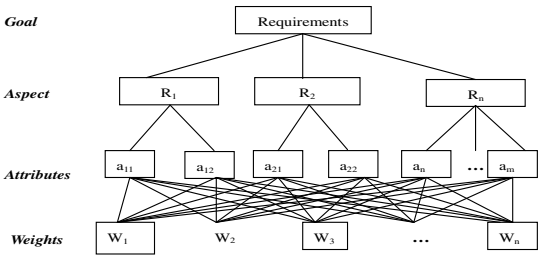


**Fig. 1.** Representation of the 4-level structure of ranking process

Let us assume that, we have $X$ and $Y$ requirements and the aim is to rank them based on the weights of attributes $a_{11},...,a_{1n}, a_{21},...,a_{2n}$ provided by the respective stakeholders, each attribute will then have to be ranked based on a weight scale. Therefore, the prioritization process consists of finding the weights that engenders the determination of relative importance of requirements. The structure describing the proposed approach is shown in Figure 2.

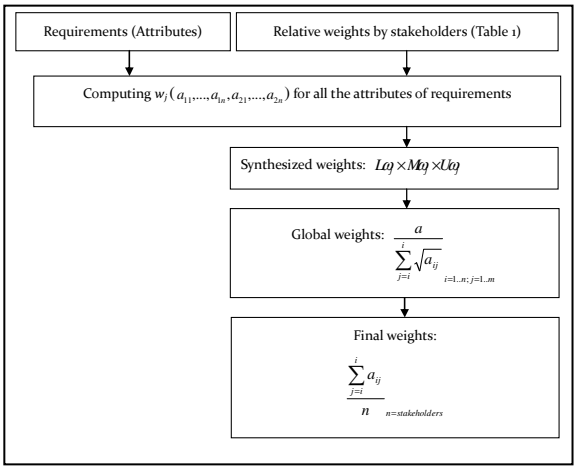


**Fig. 2.** Structure of the prioritization process

The relative value of each requirement is measured on the basis of the accrued weights across project stakeholders. The prioritized output will then be the cumulative sums of the accrued weights.

**Definition 3.3.1:** Let $X$ be a measurable requirement set that is endowed with attributes of σ-functionalities, where $N$ is all subsets of $X$. A prioritization function $g$ defined on the measurable space $(X, N)$ is a set function $g : N \to [0,1]$ which satisfies the following properties:

$$g(\phi) = 0, g(X) = 1 \tag{1}$$

But for requirement sets *X, Y;* the equation for the prioritization process will be:

$$X \subseteq Y \in N \to [0,1] \tag{2}$$

From the above definition, $X, Y, N, g$ are said to be the parameters used to measure or determine the relative weights of requirements. This process is monotonic. Consequently, the monotonicity condition is obtained as:

$$g(X \cup Y) \geq \max\{g(X), g(Y)\}; g(X \cap Y) \leq \min\{g(X), g(Y)\} \tag{3}$$

In the case where $g(X \cup Y) \geq \max\{g(X), g(Y)\}$, the prioritization function $g$ attempts to determine the total number of requirements being prioritized and if $g(X \cap Y) \leq \min\{g(X), g(Y)\}$, the function attempts to compute the relative weights of requirements provided by the relevant stakeholders.

**Definition 3.3.2:** Let $h = \sum_{i=1}^{n} X_i . 1_{X_i}$ be a simple function.

$1_{X_i}$, is the attribute function of the requirement set $X_i \in N, i = 1,...,n$; the sets $X_i$ are pairwise disjoints, but if $M(X_i)$ is the measure of the weights between all the

attributes contained in $X_i$, then the integral of *h* which is used to find the local weights of requirements is given as:

$$\int h.dM = \sum_{i=1}^{n} M(X_i).{}_{X_i} \tag{4}$$

**Definition 3.3.3:** Let $X, Y, N, g$ be the measure of weights between two sets of requirements, the integral of weights measure $g : N \rightarrow [0,1]$ with respect to a simple function *h* is defined by:

$$\int h(r)g(r) = \vee(h(r_i) \wedge g(X_i)) = \max\min\{r'_i, g(Y_i)\} \tag{5}$$

Where $h(r_i)$ is a linear combination of an attribute function $1r_i$ such that $X_1 \subset Y_1 \subset ... \subset X_n \subset Y_n$ and $X_n = \{r \mid h(r) \geq Y_n\}$. This is used to determine the global weights between requirements.

**Definition 3.3.4:** Let $X, N, g$ be a measure space. The integral of a measure of weights by the prioritization process $g : N \rightarrow [0,1]$ with respect to a simple function *h* is defined by

$$\int h(r).dg \cong \sum[(h(r_i) - h(r_{i-1})].g(X_i) \tag{6}$$

Similarly, if $Y, N, g$ is a measure space; the integral of the measure of the weights with respect to a simple function *h* will be:

$$\int h(r').dg \cong \sum[(h(r_i') - h(r'_{i-1})].g(Y_i) \tag{7}$$

However, if *g* measures the relative weights of requirements, defined on a power set *P(x)* and satisfies the definition 3.3.1 as above; the following attribute is evident:

$$\forall X, Y \in P(x), X \cap Y = \phi \Rightarrow g_2(X \cup Y) = g_2(X) + g_2(Y) + \lambda g_2(X) g_2(Y) \tag{8}$$

For $0 \leq \lambda \leq \infty$

Therefore, for requirement set $X = \{r_1, r_2, ..., r_n\}$, the density of weights measure $w_i = \{r_i\}$ can be formulated as follows:

$$w(\{r_1, r_2, ..., r_n\}) = \sum_{i=1}^{n} w_i + \lambda \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^{n} w_{i1}.w_{i2} + ... + \lambda^{n-1}.w_{i1}.w_{i2}...w_n = \frac{1}{\lambda}\langle \prod_{i=1}^{n}(1 + \lambda.w_i) - 1\rangle \tag{9}$$

For $0 \leq \lambda \leq \infty$

However, *h* is a measurable set function defined on the certain measurable space of requirement weights $X, N$ and assuming $h(r_1) \geq h(r_2) \geq ... \geq h(r_n)$, then the integral of the weights measure of requirements $g(\cdot)$ with respect to $h(\cdot)$ can be defined as follows:

$$\int h.dg = h(r_n).g(X) + [h(r_{n-1}) - h(r_n)].g(Y) + ... + [h(r'_{n-1}) - h(r'_n)] \tag{10}$$

$$= h(r_n).[g(X) - g(r_{n-1})] + h(r_{n-1}).[g(Y) - g(r'_{n-1})] + h(r'_{n-1}) \tag{11}$$

where $X = (r_1, r_2, ..., r_n); Y = (r'_1, r'_2, ..., r'_n)$.

Therefore, the computation of relative weights across all the requirements in the given sets is a dependent relation between attributes and the stakeholders. Multi-

attributes multiplicative utility function known as non-additive multi-criteria evaluation technique can be used to refine the situations that do not conform to the assumption of independence between attributes criteria [23].

In practical application of the prioritization process, $X = (r_1, r_2, ..., r_n)$; $Y = (r'_1, r'_2, ..., r'_n)$ probably represents two sets of requirements with their respective attributes that are to be ranked. In these sets, attributes are not necessary mutually independent. In order to drive the synthetic utility values, we first exploit the factor analysis technique to extract the attributes that possess common functionalities using Equation 1. This caters for requirement dependencies challenges during the prioritization process. The attributes with the same functionalities are considered to be mutually dependent. Therefore, before relative weights are assigned to the requirements by relevant stakeholders, attention should be paid to requirement dependencies issues in order to avoid redundant results.

However, when requirements evolve, it becomes necessary to add or delete from a set. The algorithm should also be able to detect this situation and update rank status of ordered requirements instantly. This is known as rank reversals. It is formally expressed as follows: (1) failure of the type $0 \rightarrow 1$ or $1 \rightarrow 0$; (2) failures of the type $0 \rightarrow \phi$ or $1 \rightarrow \phi$ (where $\phi$ = the null string) (called *deletions*); and (3) failures of the type $\phi \rightarrow 0$ or $\phi \rightarrow 1$ (called *insertions*). A weight metric *w,* on two requirement sets *(X, Y)* is defined as the smallest number of edit operations (deletions, insertions and updates) to enhance the prioritization process. Three types of rank updates operations on $X \rightarrow Y$ are defined as: a *change* operation ($X \neq \phi$ and $Y \neq \phi$), a *delete* operation ($Y = \phi$) and an *insert* operation ($X = \phi$). The weights of all the requirements can be computed by a *weight function w*. An arbitrary weight function *w* is obtained by computing all the assigned non-negative real number *w (X, Y)* on each requirement sets. This is achieved by Equations 2-7. However, in additive and non-additive measurement (rank updates) cases, Equations 8-11 is utilized to find the synthetic utilities of each attribute in the set within the same factor. On the other hand, there is mutual independence between attributes, and the measurement is an additive case, so we can utilize the additive aggregate method to conduct the synthetic utility values for all the attributes in the entire requirement sets.

Before requirements prioritization is performed, it is expected to ensure that all the attributes and requirements are mutually independent. Thereafter, the relative weights and performance score of each attribute corresponding to each requirement set is computed across all the stakeholders. Then, these scores are aggregated to obtain the final ranks of requirements. The relative weight of the *j*-th attribute is calculated by obtaining the subjective weights of stakeholders using weight scale shown in Table 1.

**Table 1.** Weight scale

| *Variables* | *Rank* | *Relative numbers* |
|---|---|---|
| Extermely important (EI) | 1 | (0.75, 0.90, 1.00) |
| More important (MI) | 2 | (0.25, 0.50, 0.65) |
| Less important (LI) | 3 | (0.15, 0.30, 0.45) |

## 4 Empirical Example

To illustrate the concept of our approach, an electronic health records system is considered in this case. A hospital would like to develop new software to replace the existing one that do not support distributed healthcare delivery services. The new system should allow a medical practitioner to administer quality healthcare from any geographical location across the three tiers of healthcare institutions (primary, secondary and tertiary). The system must be flexible enough to enable physicians gain access into the system and administer appropriate healthcare. It is required that the system be scalable and interoperable. The project consists of nine stakeholders, with four requirements sets denoted as *P*, *F*, *U*, and *M* representing Performance, Flexibility, Usability and Maintainability respectively containing fourteen attributes all together. Tables 2-4 show the relative variables, synthesized and global weights of the various attributes for the specified requirements.

**Table 2.** Relative variables

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ | $P_{15}$ | $F_{21}$ | $F_{22}$ | $F_{23}$ | $U_{31}$ | $U_{32}$ | $M_{41}$ | $M_{42}$ | $M_{43}$ | $M_{44}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | EI | EI | EI | EI | EI | MI | MI | MI | MI | MI | EI | EI | EI | EI |
| $S_2$ | EI | EI | EI | EI | EI | EI | EI | EI | EI | EI | EI | EI | EI | EI |
| $S_3$ | MI | MI | MI | MI | MI | EI | EI | EI | EI | EI | LI | LI | LI | LI |
| $S_4$ | MI | MI | MI | MI | MI | LI | LI | LI | EI | EI | EI | EI | EI | EI |
| $S_5$ | EI | EI | EI | EI | EI | MI | MI | MI | LI | LI | MI | MI | MI | MI |
| $S_6$ | MI | MI | MI | MI | MI | EI | EI | EI | LI | LI | MI | MI | MI | MI |
| $S_7$ | EI | EI | EI | EI | EI | EI | EI | EI | LI | LI | MI | MI | MI | MI |
| $S_8$ | MI | MI | MI | MI | MI | LI | LI | LI | MI | MI | MI | MI | MI | MI |
| $S_9$ | EI | EI | EI | EI | EI | MI | MI | MI | LI | LI | EI | EI | EI | EI |

**Table 3.** Synthesized weights

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ | $P_{15}$ | $F_{21}$ | $F_{22}$ | $F_{23}$ | $U_{31}$ | $U_{32}$ | $M_{41}$ | $M_{42}$ | $M_{43}$ | $M_{44}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.675 | 0.675 | 0.675 | 0.675 |
| $S_2$ | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 |
| $S_3$ | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.020 | 0.020 | 0.020 | 0.020 |
| $S_4$ | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.020 | 0.020 | 0.020 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 |
| $S_5$ | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.081 | 0.081 | 0.081 | 0.020 | 0.020 | 0.081 | 0.081 | 0.081 | 0.081 |
| $S_6$ | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.675 | 0.675 | 0.675 | 0.020 | 0.020 | 0.081 | 0.081 | 0.081 | 0.081 |
| $S_7$ | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.020 | 0.020 | 0.081 | 0.081 | 0.081 | 0.081 |
| $S_8$ | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.020 | 0.020 | 0.020 | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 | 0.081 |
| $S_9$ | 0.675 | 0.675 | 0.675 | 0.675 | 0.675 | 0.081 | 0.081 | 0.081 | 0.020 | 0.020 | 0.675 | 0.675 | 0.675 | 0.675 |

**Table 4.** Global weights

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ | $P_{15}$ | $F_{21}$ | $F_{22}$ | $F_{23}$ | $U_{31}$ | $U_{32}$ | $M_{41}$ | $M_{42}$ | $M_{43}$ | $M_{44}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.16 | 0.16 | 0.16 | 0.20 | 0.20 | 0.41 | 0.41 | 0.41 | 0.41 |
| $S_2$ | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.33 | 0.33 | 0.33 | 0.58 | 0.58 | 0.41 | 0.41 | 0.41 | 0.41 |
| $S_3$ | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 0.33 | 0.33 | 0.33 | 0.58 | 0.58 | 0.07 | 0.07 | 0.07 | 0.07 |
| $S_4$ | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 0.10 | 0.10 | 0.10 | 0.58 | 0.58 | 0.41 | 0.41 | 0.41 | 0.41 |
| $S_5$ | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.16 | 0.16 | 0.16 | 0.10 | 0.10 | 0.14 | 0.14 | 0.14 | 0.14 |
| $S_6$ | 0.13 | 0.13 | 0.40 | 0.40 | 0.40 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.14 | 0.14 | 0.14 | 0.14 |
| $S_7$ | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.14 | 0.14 | 0.14 | 0.14 |
| $S_8$ | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 0.10 | 0.10 | 0.10 | 0.20 | 0.20 | 0.14 | 0.14 | 0.14 | 0.14 |
| $S_9$ | 0.40 | 0.40 | 0.40 | 0.40 | 0.40 | 0.16 | 0.16 | 0.16 | 0.10 | 0.10 | 0.41 | 0.41 | 0.41 | 0.41 |

## 5 Experimental Results

The results displayed in Table 5 shows the summary of output executed for the prioritized requirements, obtained from preference weights of stakeholders. The overall result is shown in Table 6.

**Table 5.** Execution output

| Requirements | Stakeholders | Mean | Std. deviation |
|---|---|---|---|
| QoS ($P_{11}$) | 9 | 0.411 | 0.313 |
| Scalability ($P_{12}$) | 9 | 0.411 | 0.313 |
| Security ($P_{13}$) | 9 | 0.411 | 0.313 |
| Data communication ($P_{14}$) | 9 | 0.411 | 0.313 |
| Data redundancy ($P_{15}$) | 9 | 0.411 | 0.313 |
| Installation ease ($F_{21}$) | 9 | 0.329 | 0.329 |
| User friendly ($F_{22}$) | 9 | 0.329 | 0.329 |
| Compatibility ($F_{23}$) | 9 | 0.331 | 0.327 |
| Code change ($U_{31}$) | 9 | 0.252 | 0.318 |
| File change ($U_{32}$) | 9 | 0.252 | 0.318 |
| Documentation quality ($M_{41}$) | 9 | 0.338 | 0.320 |
| Maintenance plan ($M_{42}$) | 9 | 0.338 | 0.320 |
| Installation manual ($M_{43}$) | 9 | 0.338 | 0.320 |
| User training ($M_{44}$) | 9 | 0.338 | 0.320 |

The proposed approach has the capacity to accurately address rank reversal and dependency issues as against the existing techniques. For example, in Table 6, $R_1$ and $R_4$ emerged as prime requirements even though $R_1$ had more attributes than $R_4$. It is also applicable to large numbers of requirements. Determining the weights of stakeholder's requirements was achieved by synthesizing the priorities over all levels obtained by varying numbers of requirements.

**Table 6.** Prioritized requirements

| Requirements | Final rank |
|---|---|
| $R_1$ | 1.40 |
| $R_2$ | 0.51 |
| $R_3$ | 0.56 |
| $R_4$ | 1.00 |

## 6 Conclusion and Future Work

Many software development projects fail not because there are no skillful programmers but because there are no skillful elicitors who have the capacity of acquiring and ranking requirements in an efficient and precise manner in order to plan

for software releases. This research has proposed an approach that will help guide developers, elicitors, architects and other stakeholders in their quest to develop systems that meet the requirements of the users. Surely, when requirements are vaguely elicited, the resulting system will not function as expected even when the codes are free of errors. In conclusion, this research proposed a preference weights model for prioritizing software requirements. Four user's requirements with fourteen respective attributes were described to describe the application of the proposed approach. By using this approach, the subjective judgments can be quantified to make comparison more efficiently and reduce assessment biasness. These efforts will aid developers in designing an architecture and software with preferential requirements of stakeholders. For the future work, the implementation of the proposed approach and its application in real-world project with large number of requirements and stakeholders is underway. Also, there is need to minimize the disagreement rate between final rank weights.

**Acknowledgement.** This work is supported by the Research Management Centre (RMC) at the Universiti Teknologi Malaysia under Research University Grant (Q.J130000.2510.03H02), the Ministry of Science, Technology & Innovations Malaysia under Science Fund (R.J130000.7909.4S062) and the Ministry of Higher Education (MOHE) Under Exploratory Research Grant Scheme (R.J130000.7828.4L051).

## References

1. Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., Aurum, A.: Prioritization of quality requirements: State of practice in eleven companies. In: 19th IEEE International Requirements Engineering Conference (RE), pp. 69–78 (2011)
2. Kassel, N.W., Malloy, B.A.: An approach to automate requirements elicitation and specification. In: Proc. of the 7th IASTED International Conference on Software Engineering and Applications, Marina Del Rey, CA, USA (2003)
3. Ramzan, M., Jaffar, A., Shahid, A.: Value based intelligent requirement prioritization (VIRP): expert driven fuzzy logic based prioritization technique. International Journal of Innovative Computing 7(3), 1017–1038 (2011)
4. Aasem, M., Ramzan, M., Jaffar, A.: Analysis and optimization of software requirements prioritization techniques. In: International Conference on Information and Emerging Technologies (ICIET), pp. 1–6. IEEE (2010)
5. Tonella, P., Susi, A., Palma, F.: Interactive requirements prioritization using a genetic algorithm. Information and Software Technology 55(1), 173–187 (2013)
6. Ahl, V.: An experimental comparison of five prioritization methods. Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden (2005)
7. Berander, P., Andrews, A.: Requirements prioritization. In: Engineering and managing software requirements, pp. 69–94. Springer, Heidelberg (2005)
8. Kobayashi, A., Maekawa, M.: Need-based requirements change management. In: Proceedings. Eighth Annual IEEE International Conference and Workshop on Engineering of Computer Based Systems, pp. 171–178. IEEE (2001)

9. Karlsson, J., Wohlin, C., Regnell, B.: An evaluation of methods for prioritizing software requirements. Information and Software Technology 39(14), 939–947 (1998)
10. Duan, C., Laurent, P., Cleland-Huang, J., Kwiatkowski, C.: Towards automated requirements prioritization and triage. Requirements Engineering 14(2), 73–89 (2009)
11. Karlsson, J., Ryan, K.: A cost-value approach for prioritizing requirements. IEEE Software 14, 67–74 (1997)
12. Bebensee, T., van de Weerd, I., Brinkkemper, S.: Binary priority list for prioritizing software requirements. In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 67–78. Springer, Heidelberg (2010)
13. Perini, A., Susi, A., Avesani, P.: A Machine Learning Approach to Software Requirements Prioritization. IEEE Transactions on Software Engineering 39(4), 445–460 (2013)
14. Thakurta, R.: A framework for prioritization of quality requirements for inclusion in a software project. Software Quality Journal 21, 573–597 (2012)
15. Lima, D.C., Freitas, F., Campos, G., Souza, J.: A fuzzy approach to requirements prioritization. In: Cohen, M.B., Ó Cinnéide, M. (eds.) SSBSE 2011. LNCS, vol. 6956, pp. 64–69. Springer, Heidelberg (2011)
16. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.N.: An adaptive fuzzy decision matrix model for software requirements prioritization. In: Sobecki, J., Boonjing, V., Chittayasothorn, S. (eds.) Advanced Approaches to Intelligent Information and Database Systems. SCI, vol. 551, pp. 129–138. Springer, Heidelberg (2015)
17. Karlsson, J., Ryan, K.: A cost-value approach for prioritizing requirements. IEEE Software 14, 67–74 (1997)
18. Kukreja, N., Payyavula, S., Boehm, B., Padmanabhuni, S.: Value-based requirements prioritization: usage experiences. Procedia Computer Science 16, 806–813 (2012)
19. Kukreja, N.: Decision theoretic requirements prioritization: a two-step approach for sliding towards value realization. In: Proceedings of the 2013 International Conference on Software Engineering, pp. 1465–1467. IEEE Press (2013)
20. Ejnioui, A., Otero, C., Otero, L.: A simulation-based fuzzy multi-attribute decision making for prioritizing software requirements. In: Proceedings of the 1st Annual Conference on Research in Information Technology, pp. 37–42. ACM (2012)
21. Gaur, V., Soni, A.: An integrated approach to prioritize requirements using fuzzy decision making. IACSIT International Journal of Engineering and Technology 2(4), 320–328 (2010)
22. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.N.R.: A systematic literature review of software requirements prioritization research. Information and Software Technology 56(6), 568–585 (2014)
23. Chen, Y., Tzeng, G.: Using fuzzy integral for evaluating subjectively perceived travel costs in a traffic assignment model. European Journal of Operational Research 130(3), 653–664 (2001)