Multi-criteria group requirement prioritization in software engineering using fuzzy linguistic labels

Giovanni Daián Rottoli, Carlos Casanova

Grupo de Investigación en Inteligencia Computacional e Ingeniería de Software (GIICIS), Universidad Tecnológica Nacional Facultad Regional Concepción del Uruguay, Concepción del Uruguay, Argentina

Abstract

Requirement prioritization is a Software Engineering task that helps to choose which and in what order requirements will be implemented in each software development process iteration. In the same way, requirement prioritization is extremely useful to make decisions during iteration management. In this work a method for requirement prioritization is proposed. This method considers many experts' opinions on multiple decision criteria provided using fuzzy linguistic labels, which allows to capture the imprecision of each experts' judgment. The opinions are aggregated using a majority-guided linguistic IOWA operator considering different weights for each expert and then the requirements are prioritized considering the aggregated opinions and different weights for each evaluated dimension. The proposed method has been implemented and demonstrated using a test dataset.

Keywords

Requirement prioritization, Fuzzy logic, Linguistic labels, OWA, Software Engineering

1. Introduction

According to [1], Software Engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. Software is produced to meet the needs of its users, and the process that transforms these needs into a software is called software development process, which is one of SE's main objects of study. Several process models or life cycles have been proposed, some of them more linear from the construction point of view, while others are iterative in nature [2]. In the latter, software construction occurs simultaneously with other activities such as design and planning.

Many methodologies that utilize an iterative model (from UP's first versions to Scrum or FDD) propose to make a prioritization of the requirements, the result of which is an ordering over the requirements according to their priority level. This prioritization helps when choosing what requirements will be implemented in each iteration. In the same way, it is useful to make decisions during iteration management.

ICAIW 2021: Workshops at the Fourth International Conference on Applied Informatics 2021, October 28–30, 2021, Buenos Aires, Argentina
rottolig@frcu.utn.edu.ar (G. D. Rottoli); casanovac@frcu.utn.edu.ar (C. Casanova)
http://www.frcu.utn.edu.ar/giicis/rottolig/ (G. D. Rottoli); http://www.frcu.utn.edu.ar/giicis/casanovac/ (C. Casanova)
0000-0002-7623-2591 (G. D. Rottoli); 0000-0002-2142-2187 (C. Casanova)

The criteria taken into consideration in requirement prioritization usually varies between methodologies. For instance, in Scrum the prioritization is realized by the Product Owner by means of Product Backlog ordering [3]. Although no attribute is specified in Scrum Guide, it is expected that the ordering is performed exclusively by value (i.e., the value or importance that the correspondent functionality can bring to the client), since the Product Owner represents the client in the Development Team. In others like RUP, both technical risk and business value are taken into account [4].

However, in practice, other criteria are taken into consideration when planning an iteration. For instance, reusability may be of particular interest for allowing to save effort in future functionalities, or the capacity to be postponed if delays occurs in other tasks (a practice known as slack in XP [5]). It seems reasonable then to introduce all attributes that may be relevant to sort the requirements.

Nevertheless, the higher the number of attributes considered in the process, the less significant the comparisons are and the more difficult to perform. This phenomenon is produced by the curse of dimensionality [6], that is, more attributes introduce the impossibility of *a-priori* comparing requirements that are non-dominated (without using further preference information).

On the other hand, in the prioritization process multiple decision makers are usually involved. In OpenUP it is recommended that the whole team participate in the Iteration Planning [7]. Again, in Scrum, Sprint Planning is a ceremony in which the entire team participate and may also invite other people to attend in order to provide advice [3].

The iteration planning is, ultimately, based on the expert judgment of the development team in an unstructured decision process and is carried out through discussion and consensus among its members. Therefore, such a process has artisanal and heuristic characteristics that are far from the systematic, disciplined and quantifiable approach that defines SE. For all this, it can be concluded that the prioritization process is typically multi-criteria and multi-person.

Furthermore, an additional aspect that characterizes the prioritization process is that the valuations assigned to the attributes of each requirement suffer from the cone of uncertainty's effects [8]. That is, such valuations are only estimates of the real values of the attributes, which will become more certain as time passes. In this sense, the use of estimates in the form of numbers or classic qualitative ordinal scales constitutes a methodologically questionable approach, since it implicitly assumes that such attributes are known with certainty. Consequently, it is necessary to make use of a framework that takes into consideration the uncertainty that produces the impossibility of establishing the estimates precisely.

Fuzzy Logic [9] offers a theoretical framework that allows to capture the imprecision of expert judgments through the use of linguistic labels, which represent fuzzy sets or numbers. These labels are ordered by a fuzzy preference relationship, where a label s_i is preferable to another s_j to a certain degree that varies between 0 (s_i is not preferable at all to s_j) and 1 (s_i is at least as preferable as s_j), admitting intermediate values. These labels can be used as values of the different attributes of the requirements and obtain orders from them.

Fuzzy Logic also offers tools for decision making in contexts with multiple objectives and multiple decision makers [10], so it is judged as a highly valuable tool in solving problems such as requirement prioritization.

The rest of the paper is structured as follows. Section 2 presents the proposed method for approaching the prioritization process. Therein, subsection 2.1 introduces the Fuzzy Linguistic

Labels, 2.2 describes the relevant aggregation operators used, and 2.3 specifies the prioritization algorithm that produces the requirements ordering from the imprecise evaluations of multiple experts across multiple dimensions or attributes. Section 3 presents the case of study used to test the approach. Section 4 contrasts this proposal with related works. Finally, section 5 collects the most relevant conclusions and identifies improvements that will be addressed in future work.

2. Proposed method

2.1. Fuzzy linguistic labels

The method presented in this article uses only ordinal information through a tuple of linguistic labels sorted by their index.

Let S be a finite and totally ordered set of labels used to indicate preferences with odd cardinality where $\forall s_i, s_j \in S, s_i \geq s_j \rightarrow i \geq j$. The semantic of each linguistic term s_i is given by the ordered structure. For example, $S^1 = (Low, Middle, High)$.

One benefit of the ordinal fuzzy linguistic approach used in this method is, first, its simplicity, since this technique acts directly taking into account the order of the labels in S, and second, the fact that linguistic assessments are approximations that are handled when it is impossible or unnecessary to obtain more accurate values [11] . Thus, it is natural for the experts to provide evaluations for each requirement on each evaluated dimension using this tool, without focusing in selecting a numeric value.

Given E, a set of experts, P, a set of requirements to be prioritized, and D, a set of dimensions or objectives to be evaluated by the experts, the method must be provided with $|E|$ matrices $O^{|P|\times|D|}$, where each $O_k(l,m) \in S$ represents the opinion of expert $e_k \in E$ about the dimension $d_m \in D$ evaluated on the requirement $p_l \in P$.

Additionally, each expert $e \in E$ can be linked to an importance degree using also the ordinal scale S so their opinion will be ponderated accordingly. The same can be done with the evaluated dimensions, so each dimension $d \in D$ weighs differently in the priorization process.

2.2. Aggregation

In order to aggregate the $|E|$ aforementioned experts' opinions, an Induced Ordered Weighted Averaging Operators (IOWA) is used to generate a single general matrix with the combined opinions of all the experts. More specifically, in this work the majority guided linguistic IOWA (MLIOWA) operator proposed in [11] because it solves common problems related to the semantic of aggregated linguistic values, and it is a majority guided operator, so the weights used in the aggregation process are induced considering the majority of values in the expert importance degree vector. Nonetheless, it is important to note that any other IOWA operator with similar characteristics can be used.

First, let Neg, Max and Min be a negation operator and two comparison operators respectively, defined on a linguistic set $S = \{s_1, s_2, \ldots, s_{|S|}\}$ as follows:

$$Neg(s_i) = s_{|S|-i+1} \tag{1}$$

$$Max(s_i, s_j) = s_{max(i,j)} \tag{2}$$

$$Min(s_i, s_j) = s_{min(i,j)} \tag{3}$$

The MLIOWA operator is a function $\varphi_Q^I : (S \times S)^{|E|} \to S$, defined as follows, where I is the vector of importance related to the experts, and Q is a linguistic fuzzy quantifier representing the concept of majority in the aggregation:

$$\varphi_Q^I((I_1, p_1), \cdots, (I_{|E|}, p_{|E|})) = s_k \tag{4}$$

with $s_k \in S$ and:

$$k = round(\sum_{i=1}^{|E|} w_i \cdot ind(p_{\sigma(i)})) \tag{5}$$

such as:

1. $ind : S \to \{1, ..., |S|\}$ such that $ind(s_i) = i$
2. $\sigma : \{1, ..., |S|\} \to \{1, ..., |S|\}$ is a permutation such that $u_{\sigma(i+1)} \geq u_{\sigma(i)}, \forall i = 1, ..., |E|-1$.
3. the order inducing values u_i are calculated using the importance degrees I_i as shown in equations 6 and 7.

$$u_i = \frac{sup_i + ind(I_i)}{2} \tag{6}$$

$$sup_i = \sum_{j=1}^{|E|} sup_{ij} | sup_{ij} = \begin{cases} 1 & \text{if } |ind(I_i) - ind(I_j)| < \alpha \in \{1, ..., |S|\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

4. and, lastly, the weighting vector $\bar{w}$ calculated using the order inducing values as shown in equation 8.

$$w_i = \frac{Q\left(\frac{u_{\sigma(i)}}{|E|}\right)}{\sum_{j=1}^{|E|} Q\left(\frac{u_{\sigma(j)}}{|E|}\right)} \tag{8}$$

For further information about the MLIOWA operator see the original publication [11].

2.3. Prioritization algorithm

In a previous work, an algorithm for multi-criteria requirement prioritization was presented [12]. In this work, a variation of that work that uses only linguistic labels and includes multiple expert opinion is shown.

Given a set of linguistic labels S, a set of experts E, a set of requirements to be prioritized P, a set of dimensions or criteria D, $|E|$ matrices $O^{|P|\times|D|}$, just as was defined in 2.1, and also given an expert importance degree vector $I \in S^{|E|}$ and a criteria weight vector $\rho \in S^{|D|}$, the steps of the algorithm are the following:

First, the experts' opinions must be aggregated using the MLIOWA operator as stated in section 2.2 using as input the $|E|$ matrices O_i and the experts importance degree vector I, producing a matrix $M^{|P|\times|D|}$ with the aggregated experts' , where each $m_{ij} \in S$ represents the aggregated opinion of the set of experts about requirement $p_i \in P$ in relation to the dimension $d_j \in D$.

The next step is to build a comparison matrix for each dimension using as input the matrix M. For a specific dimension $d_j \in D$, the comparison matrix C^{d_j} represents the fuzzy preference relation $R_{d_j} \subseteq P \times P$, where $\mu_{R_{d_j}}$, defined as shown in equation 9, is the degree to which the requirement $p_i \in P$ is at least as good as $p_k \in P$ according to the criterion d_j.

$$c_{ik}^{d_j} = \mu_{R_{d_j}}(p_i, p_k) = \begin{cases} 1 & \text{if } ind(m_{ij}) \geq ind(m_{kj}) \\ 0 & \text{if } ind(m_{ij}) < ind(m_{kj}) - \beta \\ 1 - \dfrac{|ind(m_{ij}) - ind(m_{kj})|}{\beta + 1} & \text{otherwise} \end{cases} \tag{9}$$

with $\beta \in \mathbb{N}, 0 < \beta < |S|$

Then, these preference relations represented by each comparison matrix need to be weighted considering the criteria weight vector to obtain the pondered preference relations. To do this, an implication approach similar to the one presented by [13] can be used according to equation 10,

$$\mu_{R_{d_j}}^{\rho_j}(p_i, p_k) = \max(1 - \frac{ind(\rho)}{|S|}, \mu_{R_{d_j}}^{\rho_j}(p_i, p_k)) \tag{10}$$

Then, the global fuzzy preference relation R_g is the intersection

$$R_g = \bigcap_{j=1}^{|D|} R_{d_j}^{\rho_j}$$

using the classic T-norm Min as shown in equation 11.

$$\mu_{R_g}(p_i, p_k) = \min_{d_j \in D}\{\mu_{R_{d_j}}^{\rho_j}(p_i, p_k)\} \tag{11}$$

The global fuzzy preference relation is used then to build a new relation: the strict global fuzzy preference relation R_s. The complement of $\mu_{R_s}(p_i, p_k)$ (equation 12) represents the degree to which p_i does not dominate p_k, and can be used to calculate the degree to which an alternative

p_i is not strictly dominated by any other alternative. This last fuzzy relation is called fuzzy non-dominance relation and is calculated as shown in equation 13 .

$$\mu_{R_s}(p_i, p_k) = \max\{\mu_{R_g}(p_i, p_k) - \mu_{R_g}(p_k, p_i), 0\} \tag{12}$$

$$\mu_{R_{ND}}(p_i) = \min_{p_k}\{1 - \mu_{R_s}(p_i, p_k)\} = 1 - \max_{p_k}\{\mu_{R_s}(p_i, p_k)\} \tag{13}$$

The best alternative $p^{ND} \in A$ is the requirement with the highest membership degree to the fuzzy non-dominance relation R_{ND}. Also, when this value is equal to 1, the decision is said to be non-dominated and non fuzzy.

[14] proposes using the non-dominance fuzzy set to make the prioritization, but this may not be appropriate if there are alternatives that are not strictly dominated by others. For example, if there are 3 alternatives a, b and c, such that a dominates b, b dominates c and, transitively,

Algorithm 1: Proposed prioritization algorithm

Data: Linguistic label set: S
Data: Set of requirements: P
Data: Set of dimensions or criteria: D
Data: Set of experts: E
Data: $|E|$ experts' matrices: $M_i \in S^{|P| \times |D|}$
Data: Expert importance degree vector: $I \in S^{|E|}$
Data: Criteria weight vector: $\rho \in S^{|D|}$
Result: Partial order of requirements

$M =$ Aggregate experts' matrices using MLIOWA operator;
M^g = Initialize the global fuzzy preference relation matrix ;
foreach $d_j \in D$ **do**
  C^{d_j} = Build comparison matrix according to d_j using M;
  C^{d_j} = Ponderate dimension using ρ_j according to equation 10;
  $\forall i, k = \{1, ..., |P|\}, m^g_{ik} = min(m^g_{ik}, c^{d_j}_{ik})$;
end
$order = ()$;
$position = 0$;
while $P \neq \emptyset$ **do**
  $C^s =$ Build strict comparison matrix using C^g according to equation 12 ;
  ND = Build the non-dominance vector using C^s according to equation 13 ;
  p^{ND} = Get the requirements with the highest non-dominance value ;
  $position = position + 1$;
  $Order_{position} = p^{ND}$;
  Remove the requirements p^{ND} from P and from C^g;
end
Return $Order$;

a dominates c, b and c would have their membership degree equal to 0 making them equally preferable according to [14], which is not true.

In order to solve this problem, the most preferable solutions p^{ND} can be removed from the set, then the membership function of the remaining alternatives to the strict global fuzzy preference relation can be calculated again and finally their non-dominance degree. This procedure is repeated until there is no alternative left to be prioritized.

This method fits problems with either small or large numbers of requirements because it is possible to compose a single global fuzzy relation from an arbitrary number of dimensions. For this, fuzzy preference relations are used to evaluate multiple criteria with different degrees of importance in order to classify the alternatives considering the stakeholders' , weighted by their relevance. Then, a decreasing order of the solutions is obtained taking into consideration all the aforementioned aspects.

The structure of the algorithm is described in algorithm 1.

3. Case study

In order to illustrate how the proposed method works, a proof of concept using a generated dataset is shown below as a weak form of validation [15]. The dataset is comprised of 5 experts' opinions about 10 requirements related to a Content Management System (CMS) (see Table 1) evaluated on 3 different dimensions (Complexity, Reusability and Importance). The requirement set, the dimensions and the first expert evaluations were obtained from [12, 16] (see Table 2). The rest of the opinions were generated arbitrarily.

An implementation of this method is publicly available in [17], as well as all the data used for this experiment.

Firstly, in all the cases the fuzzy linguistic labels used was the following:

$$S = (s_1 : \text{“Very Low”}, s_2 : \text{“Low”}, s_3 : \text{“Middle”}, s_4 : \text{“High”}, s_5 : \text{“Very High”})$$

Additionally, the expert importance degree vector I and the criteria weight vector ρ used are the ones described below:

Table 1
Requirement set used in the proof of concept.

ID	Requirement description
1	Customize the User Interface.
2	Add Content Editor and content approver role.
3	Provide a portal to the Content provider with options.
4	Email notifications to be sent to content provider.
5	Conversion of the content to Digital format.
6	Quality check of the content.
7	Check Content format.
8	Apply security over the content.
9	Create package.
10	Create poster for advertisements.

Table 2
Example of one of the experts' opinions matrices used in the proof of concept.

Requirement	Complexity	Reusability	Importance
1	Middle	Low	High
2	Middle	Very High	Very High
3	Very High	Middle	High
4	Very Low	Very Low	Very Low
5	Middle	High	High
6	Very High	Very High	High
7	Low	Low	Low
8	Very Low	Middle	Very Low
9	Very Low	Low	Low
10	Low	Very Low	Very High

$$I = (i_1 : \text{“Middle”}, i_2 : \text{“Very High”}, i_3 : \text{“Middle”}, i_4 : \text{“Middle”}, i_5 : \text{“Middle”})$$
$$\rho = (\text{Complexity} : \text{“Very High”}, \text{Reusability} : \text{“Middle”}, \text{Importance} : \text{“Middle”})$$

Using the aforementioned input, the first step of the algorithms aims to aggregate the experts' into a single general matrix. To do this, the MLIOWA operator is used. Hereunder, the calculations for the element m_{11} are presented as an example, corresponding to the requirement 1 and the criterion "Complexity", being the experts' for this pair the following:

$$(e_1 : \text{“Middle”}, e_2 : \text{“Middle”}, e_3 : \text{“Low”}, e_4 : \text{“High”}, e_5 : \text{“Middle”})$$

The support sup_i for each expert importance degree was calculated according to equation 7, considering $\alpha = 1$, getting the vector $sup = (4, 1, 4, 4, 4)$ as result. The first element of the vector was calculated as follows, and the rest of the elements were calculated equally.

$$sup_1 = \sum_{j=1}^{5} sup_{1j} = 1 + 0 + 1 + 1 + 1 = 4$$

The next step aims to get the order inducing value vector u, calculated as follows according to the equation 6.

$$u = (\frac{4+3}{2}, \frac{1+3}{2}, \frac{4+2}{2}, \frac{4+4}{2}, \frac{4+3}{2}) = (3.5, 2, 3, 4, 3.5)$$

The vector u induces the order shown in Table 3 and allows to get the weights to weigh the experts' opinions using equation 8. In this case, the fuzzy quantifier "most of", Q, was defined by the parameters (0.3, 0.8). These weights can be seen in the same table mentioned above.

Lastly, the experts' were aggregated according to the induced order using the weights calculated before, as follows:

Table 3
Induced order and weights

σ	Expert	u	μ_Q	w_i
1	e_4	4	1	0.294
2	e_1	3.5	0.8	0.235
3	e_5	3.5	0.8	0.235
4	e_3	3	0.6	0.176
5	e_2	2	0.2	0.059

$$k = round(\sum_{i=1}^{5} w_i \cdot ind(p_{\sigma_i}))$$

$$\begin{aligned}
k =& round(0.294 \cdot ind(\text{“High”}) + 0.235 \cdot ind(\text{“Middle”}) + \\
& + 0.235 \cdot ind(\text{“Middle”}) + 0.176 \cdot ind(\text{“Low”}) + 0.059 \cdot ind(\text{“Middle”})) \\
k =& round(0.294 \cdot 4 + 0.235 \cdot 3 + 0.235 \cdot 3 + 0.176 \cdot 2 + 0.059 \cdot 3) \\
k =& round(3.115) = 3
\end{aligned}$$

Then, the aggregated value for the requirement 1 evaluated on the criterion “Complexity” is s_3 : “Middle”. The whole aggregated matrix is shown in Table 4.

After aggregating the experts’ opinions, the comparison matrices for each dimension were calculated using equation 9 (with $\beta = 1$)and then weighed using the equation 10 with the vector ρ. Then, the global fuzzy preference relation was determined as the intersection of the three aforementioned comparison matrices. The result of this process can be seen below:

Table 4
Matrix of aggregated experts

Requirement	Complexity	Reusability	Importance
1	Middle	Low	High
2	Middle	Very High	Very High
3	High	Middle	High
4	Very Low	Very Low	Very Low
5	Middle	High	High
6	Very High	Very High	High
7	Low	Low	Low
8	Very Low	Middle	Very Low
9	Very Low	Low	Low
10	Low	Very Low	Very High

Table 5
Algorithm iterations for requirements prioritization. An $\times$ shows the requirements that were excluded in the previous iterations.

Iteration / Order	Non-dominance vector	Non-dominated requirements
1	$(0.0, 0.5, 0.4, 0.0, 0.0, \mathbf{1.0}, 0.0, 0.0, 0.0, 0.4)$	$\{6\}$
2	$(0.4, \mathbf{1.0}, 0.9, 0.0, 0.5, \times, 0.0, 0.0, 0.0, 0.4)$	$\{2\}$
3	$(0.4, \times, \mathbf{1.0}, 0.0, \mathbf{1.0}, \times, 0.0, 0.0, 0.0, 0.5)$	$\{3, 5\}$
4	$(\mathbf{1.0}, \times, \times, 0.0, \times, \times, 0.4, 0.5, 0.0, \mathbf{1.0})$	$\{1, 10\}$
6	$(\times, \times, \times, 0.5, \times, \times, \times, \times, \mathbf{1.0}, \times)$	$\{9\}$
7	$(\times, \times, \times, \mathbf{1.0}, \times, \times, \times, \times, \times, \times)$	$\{4\}$

$$M^g = C^{\text{Complexity}} \cap C^{\text{Reusability}} \cap C^{\text{Importance}} = \begin{pmatrix} 1 & .4 & .5 & 1 & .4 & 0 & 1 & .5 & 1 & .5 \\ 1 & 1 & .5 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & .4 & 1 & 1 & .5 & .4 & 1 & 1 & 1 & .5 \\ 0 & 0 & 0 & 1 & 0 & 0 & .5 & .4 & .5 & .4 \\ 1 & .5 & .5 & 1 & 1 & 0 & 1 & 1 & 1 & .5 \\ 1 & .5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & .5 \\ .4 & .4 & 0 & 1 & .4 & 0 & 1 & .5 & 1 & .4 \\ 0 & 0 & 0 & 1 & 0 & 0 & .5 & 1 & .5 & .4 \\ 0 & 0 & 0 & 1 & 0 & 0 & .5 & .5 & 1 & .4 \\ .5 & .4 & 0 & 1 & .4 & 0 & .5 & .4 & .5 & 1 \end{pmatrix}$$

Lastly, the strict comparison matrix was computed and then the partial order was generated by iterative calculating the non-dominance vector and selecting those requirements with the biggest non-dominance membership value. The non-dominance vector calculated in each iteration as well as the non-dominated requirements are shown in Table 5.

$$C^s = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & .6 & .5 & 1 & .0 \\ .6 & 0 & .1 & 1 & .5 & 0 & .6 & 1 & 1 & .6 \\ .5 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .6 & 0 & 0 & 1 & 0 & 0 & .6 & 1 & 1 & .1 \\ 1 & .5 & .6 & 1 & 1 & 0 & 1 & 1 & 1 & .5 \\ 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & .6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .6 & 0 & 0 & .1 & 0 & .1 & 0 \end{pmatrix}$$

The results are consistent with the results obtained in [12]: when the of a single expert are provided, the two methods are equivalent.

4. Related work

According to [18], only a small number of articles uses fuzzy logic for requirement prioritization in software engineering. These studies differ from our proposal as describe here under.

[19] suggested a framework that uses fuzzy linguistic terms parameterized using fuzzy numbers. In contrast, our methods do not need to use fuzzy numbers due to the fact that the semantics of each label is associated to its order in the fuzzy linguistic label set. Additionally, our method allows to integrate multiples , which is not considered in the previously mentioned proposal. This also occurs in [20]. The authors designed a method for requirement prioritization easy to use and implement but do not consider multiple experts' .

On the other hand, [21] considers the opinion of multiple stakeholders parameterized with triangular fuzzy numbers as in [19] but do not consider multiple objectives or criteria as our method does.

Then, [22] proposes a method for fusing multiples orders of priority, given by multiple experts with different degrees of importance. This method allows the stakeholders not to include into their order all the requirements, which is a tremendous advantage if the experts do not have the same degree of expertise on the different evaluated criteria. However, this method uses as input an order of requirements per expert, which is not the case of our algorithm that uses the experts' opinions about the requirements instead.

Lastly, [23] focuses on predicting changes in requirements priority, which is especially useful in software engineering groups that use agile methodologies.

To sum up, the algorithm proposed in this article allows the user to generate a partial order of a set of requirement based on multiple experts' opinions with different importance degrees on multiple criteria, that also are weighed. Moreover, the proposed method takes as input fuzzy evaluations using fuzzy linguistic labels, which allows the experts to provide their opinions in a more familiar way. The semantics of the linguistic labels is given by their position in the ordered fuzzy linguistic label set, so they do not depend on the definition of fuzzy numbers.

The method is easy to implement and use. This can be seen in the implementation provided in [17], making it highly applicable in the industry, which is a common limitation remarked in [18].

5. Conclusion and future work

This work presents a new method for requirement prioritization that uses as input the opinions of many experts on many decision criteria, expressed using fuzzy linguistic labels. The opinions are aggregated using a majority guided linguistic IOWA considering weights for each expert, and then, the requirements are compared based on the aggregated experts' opinions on the evaluated dimension, which are also weighed by their importance. Moreover, the weights linked to the criteria and to the experts are expressed using also fuzzy linguistic labels.

The proposed method was demonstrated using a case of study that works as a weak form of validation. The algorithm was implemented in such a way that the user only has to provide the opinions and basic configurations to use it. This is a very desirable property, but a more rigorous empirical study is needed to determine how the proposal works in a real context with real users.

Finally, it is important to remark that some improvements can be made, including allowing the experts not to give an opinion for all the evaluated dimensions, considering consensus metrics in the prioritization process, using and comparing other IOWA operators, using and comparing others T-norms such as Lukasiewicz's T-norm, and using a different linguistic label set for each dimension.

Acknowledgments

Thanks to the National Technological University from Argentina (UTN) through the project *SIU-*

TICU0005297TC: Preference-based Multi-objective Optimization Approaches applied to Software Engineering, and to Tomás Casanova and Kevin-Mark Bozell Poudereux, for proofreading the translation.

References

[1] ISO, Iso/iec/ieee international standard - systems and software engineering–vocabulary, ISO/IEC/IEEE 24765:2017(E) 1 (2017) 1–541. doi:10.1109/IEEESTD.2017.8016712.

[2] P. Bourque, R. E. Fairley (Eds.), SWEBOK: Guide to the Software Engineering Body of Knowledge, version 3.0 ed., IEEE Computer Society, 2014. URL: http://www.swebok.org/.

[3] K. Schwaber, J. Sutherland, The scrum guide, Scrum Alliance 21 (2011) 1.

[4] P. Kroll, P. Kruchten, The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP, Addison-Wesley Professional, 2003.

[5] K. Beck, C. Andres, Extreme programming explained: Embrace change. 2-nd edition, 2004.

[6] R. E. Bellman, S. E. Dreyfus, Applied dynamic programming, Princeton university press, 2015.

[7] M. Cossentino, V. Hilaire, V. Seidita, The openup process, in: Handbook on Agent-Oriented Design Processes, Springer, 2013, pp. 491–566.

[8] S. McConnell, Software project survival guide, Pearson Education, 1998.

[9] L. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353. doi:10.1016/S0019-9958(65)90241-X.

[10] G. J. Klir, B. Yuan, Fuzzy sets and fuzzy logic: theory and applications, Possibility Theory versus Probab. Theory 32 (1996) 207–208.

[11] E. Herrera-Viedma, G. Pasi, A. G. Lopez-Herrera, C. Porcel, Evaluating the information quality of web sites: A methodology based on fuzzy computing with words, Journal of the American Society for Information Science and Technology 57 (2006) 538–549. doi:10.1002/asi.20308.

[12] M. L. Gabioud, C. Casanova, Priorización en ingeniería de requerimientos con preferencias difusas, in: VII Congreso Nacional de Ingeniería Informática/Sistemas de Información, 2019, p. 1.

[13] R. R. Yager, A new methodology for ordinal multiobjective decisions based on fuzzy sets, in: D. Dubois, H. Prade, R. R. Yager (Eds.), Readings in Fuzzy Sets for Intelligent Systems, Morgan Kaufmann, 1993, pp. 751–756. doi:10.1016/B978-1-4832-1450-4.50080-8.

[14] H. Borzęcka, Multi-criteria decision making using fuzzy preference relations, Operations Research and Decisions 22 (2012).

[15] R. Wieringa, Design science methodology for information systems and software engineering, Springer, 2014. doi:10.1007/978-3-662-43839-8.

[16] M. Muqeem, M. R. Beg, A fuzzy based approach for early requirement prioritization, International Journal Of Computers & Technology 15 (2015) 6480–6490.

[17] G. D. Rottoli, Gdrottoli/requirementprioritization: Rp v1.1.0, 2021. doi:10.5281/zenodo.5327643.

[18] F. A. Bukhsh, Z. A. Bukhsh, M. Daneva, A systematic literature review on requirement

prioritization techniques and their empirical evaluation, Computer Standards & Interfaces 69 (2020) 103389. doi:10.1016/j.csi.2019.103389.

[19] D. C. Lima, F. Freitas, G. Campos, J. Souza, A fuzzy approach to requirements prioritization, in: M. B. Cohen, M. Ó Cinnéide (Eds.), Search Based Software Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 64–69. doi:10.1007/978-3-642-23716-4_8.

[20] A. Ejnioui, C. Otero, L. Otero, A simulation-based fuzzy multi-attribute decision making for prioritizing software requirements, in: Proceedings of the 1st Annual Conference on Research in Information Technology, RIIT '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 37–42. doi:10.1145/2380790.2380800.

[21] P. Achimugu, A. Selamat, R. Ibrahim, Using the fuzzy multi-criteria decision making approach for software requirements prioritization, Jurnal Teknologi 77 (2015). doi:10.11113/jt.v77.6321.

[22] F. Franceschini, D. Maisano, L. Mastrogiacomo, Customer requirement prioritization on qfd: a new proposal based on the generalized yager's algorithm, Research in Engineering Design 26 (2015) 171–187. doi:10.1007/s00163-015-0191-2.

[23] A. Sapunkov, T. Afanasieva, Software for automation of user requirements prioritization, in: Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis, ICGDA 2019, Association for Computing Machinery, New York, NY, USA, 2019, p. 1–5. doi:10.1145/3318236.3318251.