Neutrosophy for software requirement prioritization

Ronald Barriga Dias¹, Wilmer Ortiz Choez², Inelda Martillo Alcivar³, Wilber Ortiz Aguilar⁴

¹Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Guayaquil Ecuador. E-mail: ronald.barrigad@ug.edu.ec
²Universidad de Guayaquil, Facultad de Comunicación Social, Guayaquil Ecuador. E-mail: wilmer.ortizc@ug.edu.ec
³Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Guayaquil Ecuador. E-mail: inelda.martilloa@ug.edu.ec
⁴Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Guayaquil Ecuador. E-mail: Wilber.ortiza@ug.edu.ec

Abstract
Software engineers are involved in complex decisions that require multiple viewpoints. A specific case is the requirement prioritization process. This process is used to decide which software requirement to develop in certain release from a group of candidate requirements. Criteria involved in this process can involve indeterminacy. In this paper a software requirement prioritization model is developed based on SVN numbers. Finally, an illustrative example is presented in order to show the proposed model.

Keywords: requirement engineering, software requirement prioritization, SVN numbers.

1. Introduction
Software quality is influenced by the ability to satisfy client and user needs obtained and described in software requirements [1]. Many models have been proposed for software requirement prioritization [1-7]. However, these proposals present limitations for dealing with indeterminacy.

In order to overcome the drawbacks identified, in this contribution we propose a novel requirement prioritization process based on SVN numbers.

In software requirement prioritization intervene different stakeholders approaching to the decision problem from different points of view. It is moreover a multidimensional problem dealing with multiple criteria of diverse nature [8]. Therefore, the proposed model is based on a decision analysis scheme [9] and the approach presented in [8]. In order to deal with heterogeneous information provided by several experts.

This paper is structured as follows: Section 2 outlines a scheme of software prioritization. Section 3 shows the theory of neutrosophy. Section 4 presents our framework for software requirements prioritization. Section 5 shows an illustrative example of the proposed model. The paper ends with conclusions and further work recommendations in Section 6.

2. Software requirement prioritization.
One frequent reason that causes low quality software is associated to problems related to identifying and selecting the most important requirements [10]. Software requirement prioritization can be modeled like a decision making problem, making it suitable to a decision analysis scheme[9]. Decision analysis is a discipline whose purpose is to help decision maker to reach a consistent decision [11].

Our proposal for a software requirement prioritization model dealing with indeterminacy is based on the classical decision analysis scheme. In this paper the software requirement prioritization process is modeled as a type of a Multi-Expert Multi-Criteria decision making problem due to the complexity of the problem where multiple criteria and experts are involved [10, 12].

In the software requirement prioritization process, it is very difficult to express reality in a quantitative way. Fuzzy set theory, introduced by Zadeh[13] in 1965, offers a mathematical model to deal with this kind of uncertainty. The fuzzy linguistic approach is based in the fuzzy set theory and especially in linguistic variable concept [14, 15]. This fact is important in software requirement prioritization where evaluation results are used to make decisions by software engineers in high complexity environment [16]. Current process of software prioritization don’t deal with indeterminacy.

3. Neutrosophy
Neutrosophy [17] is a philosophy branch developed for dealing with indeterminacy (Figure 2). Neutrosophy have been the base for developing new methods to handle indeterminate and inconsistent information like neutrosophic sets an neutrosophic logic [18, 19].
Fig. 1. Static context of Neutrosophic logic [20].

The truth value in neutrosophic set is as follows [21]:
Let \( N \) be a set defined as: \( N = \{ (T, I, F) : T, I, F \subseteq [0, 1] \} \), a neutrosophic valuation \( n \) is a mapping from the set of propositional formulas to \( N \), that is for each sentence \( p \) we have \( v(p) = (T, I, F) \).

Single valued neutrosophic set (SVNS ) [22] was developed with the goal of facilitate the real applications of neutrosophic set and set-theoretic operators.

A single valued neutrosophic set (SVNS) has been defined as follows [22]:
Let \( X \) be a universe of discourse. A single valued neutrosophic set \( A \) over \( X \) is an object having the form:
\[
A = \{ x, uA(x), rA(x), vA(x) : x \in X \}
\]
where \( uA(x) : X \rightarrow [0,1], rA(x) : X \rightarrow [0,1] \) and \( vA(x) : X \rightarrow [0,1] \) with \( 0 \leq uA(x) + rA(x) + vA(x) \leq 3 \) for all \( x \in X \). The intervals \( uA(x) \), \( rA(x) \) and \( vA(x) \) denote the truth - membership degree, the indeterminacy-membership degree and the falsity membership degree of \( x \) to \( A \), respectively.

Single valued neutrosophic numbers (SVN number) is denoted by \( A=(a,b,c) \), where \( a, b, c \in [0,1] \) and \( a+b+c \leq 3 \).

4. A software requirement prioritization model

Our aim is develop a software requirement prioritization model based on the linguistic decision analysis schema that can deal with criteria evaluated with SVN numbers. The model consists of the following phases (graphically, Figure 2):

1. Evaluation framework:
In this phase, the evaluation framework is defined to fix the requirement prioritization problem structure. The framework is established as follows:
   - Let \( E=\{e_1, e_2, ..., e_n\} (n \geq 2) \) be a set of experts.
   - Let \( C=\{c_1, c_2, ..., c_k\} (k \geq 2) \) be a set of criteria.
   - Let \( R=\{r_1, r_2, ..., r_m\} (m \geq 2) \) be a set of requirements.

   Each expert can use SVN numbers to asses each criteria, attending to its nature.

2. Gathering information:
Once the framework has been defined, the knowledge of the set of experts must be obtained. Each expert provides their preferences by using utility vectors. The utility vector [23] is represented in the following way:
   - \( p^j_i = \{p^1_i, p^2_i, ..., p^l_j\} \). Where \( p^l_j \) is the preference provided to the criterion \( c_k \) of the requirement \( r_i \) by the expert \( e_i \).

3. Rating software requirements.
The aim of this phase is to obtain a collective linguistic global assessment easily interpretable for software engineers. To do so the information is unified and aggregated.
Finally those more prioritized are identified. This phase in based the approach reviewed in the Section 3.

A two-step aggregation process is developed with the aim of compute a global evaluation of each software requirement.

We obtain for each expert an assessment for each requirement.

The final aim of the ranking process is to obtain a global evaluation of each requirement according to all experts. To do so, this process will aggregate all the experts’ collective assessment. In decision analysis schema aggregation operator are important for rating options. Some aggregation operators have been proposed for SVN numbers [17, 24]. Single valued neutrosophic weighted averaging (SVNWA) aggregation operator was proposed by Ye [24] for SVNSs as follows[25]:

\[
F_w(A_1, A_2, \ldots, A_n) = \left(1 - T_{A_j}(x)\right)^{w_j},
\]

\[
\langle 1 - \prod_{j=1}^{n} (l_{A_j}(x)) \rangle = \left(1 - \prod_{j=1}^{n} (u_{A_j}(x)) \right)^{w_j}, \quad (2)
\]

\[
\prod_{j=1}^{n} (F_{A_j}(x))^{w_j}
\]

We propose this operator to establish different weights for each expert, taking into account their knowledge and their significance in software prioritization process.

**Rating of the requirements**

The final step in the prioritization process is to establish a ranking among software requirements, this ranking allows selecting the requirements with more value and postponing or rejecting the development of others making more effective the software development process.

For rating alternatives an ideal option is constructed [26, 27]. The evaluation criteria can be categorized into two categories, benefit and cost. Let \( C^+ \) be a collection of benefit criteria and \( C^- \) be a collection of cost criteria. The ideal alternative is defined as:

\[
I = \left\{ \begin{array}{l}
\max_{k=1}^{n} U_j / u \in C^+, \min_{k=1}^{n} U_j / u \in C^- \\
\min_{k=1}^{n} I_j / i \in C^+, \max_{k=1}^{n} I_j / i \in C^- \\
\end{array} \right\}
\]

Alternatives are rating according Euclidean distance to \( I \) (2). Ranking is based in the global distance to the ideal. If alternative \( x_i \) is closer to \( I \) the distance measure (\( s_i \) closer) is better is the alternative [28].

Alternatives could be rated according Euclidean distance in SVN [26, 29].

Let \( A^* = (A_1^*, A_2^*, \ldots, A_n^*) \) be a vector of \( n \) SVN numbers such that \( A_j^* = (a_{ij}^*, b_{ij}^*, c_{ij}^*) \) (where \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, m \) be \( m \) vectors of SVN numbers such that \( B_{ij} = (b_{ij}, c_{ij}, e_{ij}) \) (where \( i = 1, 2, \ldots, m \), \( j = 1, 2, \ldots, n \) ). Then the separation measure between \( B_i \)'s \( y \) is defined as follows:

\[
s_i = \left( \frac{1}{n} \sum_{j=1}^{n} \left( (|a_{ij} - a_{ij}^*|^2 + |b_{ij} - b_{ij}^*|^2 + |c_{ij} - c_{ij}^*|^2) \right)^{1/2} \right)
\]

The best requirement is the one with the mimimum distance to ideal.

**5. Illustrative Example**

In this section, we present an illustrative example in order to shown the applicability of the proposed model.

**A. Evaluation framework**

In this case study the evaluation framework is compose by:

- 3 experts E={\( e_1, e_2, e_3 \)}, who evaluate 3 requirements R={\( r_1, r_2, r_3 \)}, where are involved 5 criteria C={\( c_1, c_2, \ldots, c_5 \)} which are shown below:

- \( c_1 \): Importance for the customers
- \( c_2 \): Value
- \( c_3 \): Cost
- \( c_4 \): Technical Complexity
- \( c_5 \): Risks

The following linguistic terms are used (Table I).

<table>
<thead>
<tr>
<th>Linguistic terms</th>
<th>SVNSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extremely good (EG)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>Very good (VG)</td>
<td>(0.9,0.1,0.1)</td>
</tr>
<tr>
<td>Good (G)</td>
<td>(0.8,0.15,0.20)</td>
</tr>
<tr>
<td>Medium (M)</td>
<td>(0.70,0.25,0.30)</td>
</tr>
<tr>
<td>Medium bad (MB)</td>
<td>(0.60,0.35,0.40)</td>
</tr>
<tr>
<td>Bad (B)</td>
<td>(0.50,0.50,0.50)</td>
</tr>
<tr>
<td>Very bad (VB)</td>
<td>(0.40,0.65,0.60)</td>
</tr>
<tr>
<td>Very very bad (VVB)</td>
<td>(0.30,0.75,0.70)</td>
</tr>
<tr>
<td>Extremely bad (EB)</td>
<td>(0.20,0.85,0.80)</td>
</tr>
<tr>
<td>Extremely bad (EB)</td>
<td>(0.10,0.90,0.90)</td>
</tr>
<tr>
<td>Extremely bad (EB)</td>
<td>(0,1,1)</td>
</tr>
</tbody>
</table>

**B. Gathering information**

Once the evaluation framework has been determined the information about therequirements is gathered (see Table II).
Table II. An illustrative example of gathering information

<table>
<thead>
<tr>
<th></th>
<th>e₁</th>
<th>e₂</th>
<th>e₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>₁</td>
<td>VV</td>
<td>VG</td>
<td>EG</td>
</tr>
<tr>
<td>₂</td>
<td>M</td>
<td>G</td>
<td>MB</td>
</tr>
<tr>
<td>₃</td>
<td>VG</td>
<td>M</td>
<td>VG</td>
</tr>
<tr>
<td>₄</td>
<td>G</td>
<td>M</td>
<td>VG</td>
</tr>
<tr>
<td>₅</td>
<td>M</td>
<td>G</td>
<td>VG</td>
</tr>
</tbody>
</table>

C. Rating Requirements

In this example, is applied a two-step aggregation process to compute a collective evaluation for software requirements. In our problem the SVNWA is used to aggregate evaluations by requirement for each expert. In this case the weighting vectors to compute the collective evaluation is \( V = (0.3, 0.3, 0.4) \).

Table III. An illustrative example of unified and aggregated information

<table>
<thead>
<tr>
<th></th>
<th>r₁</th>
<th>r₂</th>
<th>r₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>₁</td>
<td>(0.24, 0.2, 0.12)</td>
<td>(0.18, 0.18, 0.14)</td>
<td>(0.19, 0.0, 0.0)</td>
</tr>
<tr>
<td>₂</td>
<td>(0.41, 0.44, 0.35)</td>
<td>(0.32, 0.3, 0.25)</td>
<td>(0.46, 0.44, 0.35)</td>
</tr>
<tr>
<td>₃</td>
<td>(0.38, 0.0, 0.17)</td>
<td>(0.29, 0.27, 0.19)</td>
<td>(0.54, 0.61, 0.5)</td>
</tr>
<tr>
<td>₄</td>
<td>(0.21, 0.21, 0.17)</td>
<td>(0.49, 0.49, 0.41)</td>
<td>(0.21, 0.21, 0.17)</td>
</tr>
<tr>
<td>₅</td>
<td>(0.49, 0.49, 0.41)</td>
<td>(0.24, 0.25, 0.2)</td>
<td>(0.26, 0.23, 0.16)</td>
</tr>
</tbody>
</table>

From this information, the ideal alternative is calculated (Table IV).

Table IV. Ideal alternative

<table>
<thead>
<tr>
<th></th>
<th>( E^+ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>₁</td>
<td>(0.2,0,0)</td>
</tr>
<tr>
<td>₂</td>
<td>(0.4,0.3,0.25)</td>
</tr>
<tr>
<td>₃</td>
<td>(0.38, 0.61, 0.5)</td>
</tr>
<tr>
<td>₄</td>
<td>(0.49,0.21,0.17)</td>
</tr>
<tr>
<td>₅</td>
<td>(0.24,0.49,0.41)</td>
</tr>
</tbody>
</table>

The results of the calculation of the distances allow requirement.

Table V. Distance to ideal alternative

<table>
<thead>
<tr>
<th></th>
<th>r₁</th>
<th>r₂</th>
<th>r₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>₁</td>
<td>0.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>₂</td>
<td>0.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>₃</td>
<td>0.45</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finally, we put in order all collective evaluations and we establish a ranking among requirements with the purpose...
of identifying the best ones. In the case study the ranking is as follow: $r_1 > r_2 > r_3$

After application in this case study the model is found to be practical to use. The aggregation process gives a high flexibility so the model can be adapted to different situations.

6. Conclusions

In this paper, we have proposed a prioritization model based on the decision analysis scheme that can manage SVN numbers. We have applied the proposed model to an illustrative example. The model was found to be flexible and practical to use. The developing of software tool to automate the model is an area of future work.

References


based on fuzzy cognitive maps and OWA operators. in Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on. 2014. IEEE.


37 Topal, S. and Smaradache, F. A Lattice-Theoretic Look: A Negated Approach to Adjectival (Intersective, Neutrosophic and Private) Phrases. The 2017 IEEE International Conference on Innovations in Intelligent SysTem and Applications (INISTA 2017); (accepted for publication).


Received: August 10, 2017. Accepted: August 31, 2017.