

A Fuzzy Approach to Prioritisation of Security Requirements in a Constrained Agile Software Development Environment

Nalindren Naicker

Durban University of Technology,
Durban, South Africa

Manoj Maharaj

University of KwaZulu-Natal
Durban, South Africa

Abstract

Requirements Engineering (RE) is software engineering process that takes place early namely, during the planning phases of software development. The spate of hacking incidents, more especially the ransomware infections, recently has suggested a more robust approach to web application security during RE. This will not only prevent these vulnerabilities but also the necessary rework after a hacking incident. The prioritisation process in RE is instrumental in ensuring if a security requirement gets implemented or is kept on hold indefinitely. A desktop literature review revealed ample scope for fuzzy TOPSIS as a security requirements prioritisation technique. The aim of this research study was to assess the viability of a new and more inclusive technique of more precisely ranking security requirements. A fuzzy automated tool was developed to test this new approach to ranking security requirements. The method will ensure the implementation of the most important security requirements to secure the system. The Design Science Research Methodology guided the development of the automated fuzzy software tool. The automated tool was evaluated in a qualitative study at 17 software development companies. The researcher used structured interviews and document reviews as the primary research instruments. Qualitative data was analyzed deductively using content analysis. It was found that an immediate benefit of the tool is that it prevented biases and autocratic leaders from influencing decision making during security risk analysis. The study concluded that the automated fuzzy tool showed positive results for ranking security requirements in Agile RE. Further research on the usability of the software tool is recommended.

Keywords: Agile, Security, Requirements Engineering, Fuzzy TOPSIS.

Introduction

In real life decision makers very often are required to make important choices from several alternatives. The selection of an incorrect alternative based on subjective human judgement can have negative consequences. In software development for example the incorrect decision of a project manager on the choice of what features to implement can impact the security of applications. A more precise way at arriving at decisions involving subjective human preferences is required. Multi-criteria decision making (MCDM) problems integrated with fuzzy set theory provides a suitable solution to arrive at a precise judgement based on imprecise information (Lima et al. 2014). This approach is investigated in pursuit of a more rigorous method to prioritise security requirements in a non-prescriptive software development methodology such as Agile Software Development (ASD).

Research Methodology

A research gap was identified for an automated requirements prioritisation tool to manage the requirements prioritisation process. A desktop literature review into the various requirements prioritisation techniques revealed that a gap existed for the utilization of fuzzy TOPSIS as a prioritisation technique (Achimugu, Selamat, Ibrahim, & Mahrin, 2014). Literature has highlighted a shortage of automated tools to support secure RE processes (Souag, Mazo, Salinesi, & Comyn-Wattiau, 2015). Therefore, the development of an automated tool using the fuzzy TOPSIS approach to support secure RE activities in Agile Software Development will contribute to the existing body of knowledge and fill a research gap.

Participants

Developers from seventeen software development companies were interviewed based on quota sampling to evaluate the automated fuzzy tool. The composition of the companies in terms of their specialisations were as follows: Gaming (5), Medical (3); Transport (1); Finance (2); Education (2) and Non-Specific (4). The tool was constructed from the fuzzy TOPSIS method based on fuzzy set theory.

Procedure

The interview schedule was constructed through a process of synthesis from the evaluation frameworks in the literature on requirements prioritisation techniques. Questions were carefully crafted to evaluate the fuzzy TOPSIS technique for requirements prioritisation. Recruitment of participants took place at the Company. Before the data collection session, signed consent forms were obtained from participants. The rights of all respondents were respected by treating them equally with dignity and respect. The freedom of choice of the individual was respected, for example they were allowed to discontinue with the interviews at any time, participants were free to express concerns at any time and they were given access to any information that they required. Participants were provided with repeated assurance of anonymity. Interviews in this study were audio recorded. The researcher ensured that the recording was audible for the transcriber and sufficient

time was allocated for transcribing. Qualitative data was analysed using content analysis. Coding was a way of indexing or categorizing the text in order to establish a framework of thematic ideas about it. A computer program, Nvivo Pro was used for filing and retrieving coded information.

Fuzzy TOPSIS method

Fuzzy TOPSIS is a method used when a group decision is required for the selection of an alternative from a number of alternatives (possible choices) using multiple decision making criteria to rate alternatives (Martin, 2011). Normally a linguistic rating scale such as *very poor*, *poor*, *satisfactory*, *good* and *very good* are used to rate alternatives based on set criteria. The linguistic rating, for example *good* is regarded as vague and imprecise. It is difficult to distinguish which alternative is better from a choice of two alternatives when both are given a rating of *good*. The vague rating *good* can be easily represented as a triangular fuzzy number using fuzzy set theory, for example using a numeric scale of 0-9, *good* represented as a triangular fuzzy number can be written as (7, 8, 9). In the fuzzy TOPSIS approach linguistic ratings are replaced by triangular fuzzy numbers provide mathematically precise ways of arriving at decisions involving human judgement (Sodhi, 2012). The final judgement integrates the expert opinion of all decision makers.

In software development, the choice of a security requirement (security feature of the system) for development from the product backlog (store) requires multiple decision makers who rate requirements based on multiple criteria from a collection of requirements (alternatives). Incorrect decision making can result in low customer satisfaction, losses and hence project failure.

The fuzzy TOPSIS algorithm comprises of the following steps, according to (Sodhi, 2012):

Step 1:

Decision makers must determine the linguistic and fuzzy weighting of evaluation criteria and alternatives.

Step 2:

(i) Aggregate weights of each criterion as follows:

Let $K = \text{number of decision makers}$

Let $j = 1, 2, \dots, n$ representing the criteria

$$\tilde{w}_j^k = (w_{j1}, w_{j2}, w_{j3})$$

$$\text{where: } w_{j1} = \min_k \{w_{jk1}\}, w_{j2} = \frac{1}{K} \sum_{k=1}^K w_{jk2}, w_{j3} = \max_k \{w_{jk3}\} \quad (1)$$

(ii) Aggregate weights of each alternative as follows:

Let $k = \text{number of decision makers}$

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

Aggregate weightings $\tilde{x}_{ij} = (l_{ij}, m_{ij}, u_{ij})$ of alternatives (i) with respect to each criterion (j) based on the fuzzy ratings by the k decision maker is expressed by the calculations below:

$$l_{ij} = \min_k \{l_{ij}^k\}, m_{ij} = \frac{1}{K} \sum_{k=1}^K m_{ij}^k, u_{ij} = \max_k \{u_{ij}^k\} \quad (2)$$

(iii) Aggregate fuzzy decision matrix for the alternatives (\tilde{D}) is expressed below:

$$\tilde{D} = \begin{matrix} & C_1 & C_2 & C_j & C_m \\ A_1 & \tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{1j} & \tilde{x}_{1m} \\ A_i & \vdots & \vdots & \vdots & \vdots \\ A_n & \tilde{x}_{n1} & \tilde{x}_{n2} & \tilde{x}_{nj} & \tilde{x}_{nm} \end{matrix} \quad (3)$$

Step 3:

The normalized fuzzy decision matrix (\tilde{R}) is written as follows:

Let \tilde{R} = normalized decision matrix

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

Let (l_{ij}, m_{ij}, u_{ij}) = aggregated weighting of alternative (i) with respect to criterion (j)

Let \tilde{r}_{ij} = triangular fuzzy number of alternative (i) with respect to criterion (j) in matrix r with n rows and s columns

The normalized decision matrix is calculated as follows:

$$(\tilde{R}) = [\tilde{r}_{ij}]_{n \times s}$$

$$\text{where } \tilde{r}_{ij} = \left(\frac{l_{ij}}{u_j^+}, \frac{m_{ij}}{u_j^+}, \frac{u_{ij}}{u_j^+} \right) \text{ and } u_j^+ = \max_i u_{ij} \quad (\text{benefit criteria}) \quad (4)$$

therefore $u_j^+ = \max_i (9, 9, 9)$

Step 4:

The weighted normalized decision matrix, (\tilde{V}) is computed as follows:

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

The weights of the evaluation criteria, (\tilde{w}_j) is multiplied by the elements \tilde{r}_{ij} of the normalized fuzzy decision matrix.

$$(\tilde{V}) = [\tilde{v}_{ij}]_{n \times s} \quad (5)$$

where $\tilde{v}_{ij} = \tilde{r}_{ij} \times \tilde{w}_j$

Step 5:

The Fuzzy Positive Ideal Solution (FPIS, A^+) and the Fuzzy Negative Ideal Solution (FNIS, A^-) is calculated as follows:

Let $j = 1, 2, \dots, n$ represent the criteria

$$A^+ = \{ \tilde{v}_1^+, \tilde{v}_j^+, \dots, \tilde{v}_n^+ \} \quad (6)$$

$$A^- = \{ \tilde{v}_1^-, \tilde{v}_j^-, \dots, \tilde{v}_n^- \} \quad (7)$$

where $\tilde{v}_j^+ = (1, 1, 1)$ and $\tilde{v}_j^- = (0, 0, 0)$

Step 6:

The distances d_j^+ and d_j^- of each alternative from respectively \tilde{v}_j^+ and \tilde{v}_j^- are calculated as follows:

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

$$d_i^+ = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^+) \quad (8)$$

$$d_i^- = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^-) \quad (9)$$

Where d represents the distance between two triangular fuzzy numbers. For example the distance $d(\tilde{v}_{ij}, \tilde{v}_j^+)$ where $\tilde{v}_{ij} = (l_1, m_1, u_1)$ and $\tilde{v}_j^+ = (l_2, m_2, u_2)$ is expressed as follows:

$$d(\tilde{v}_{ij}, \tilde{v}_j^+) = \sqrt{\frac{1}{3} [(l_1 - l_2)^2 + (m_1 - m_2)^2 + (u_1 - u_2)^2]} \quad (10)$$

Step 7:

The closeness coefficient (CC_i) is calculated as follows:

$$CC_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (11)$$

Step 8:

Define the ranking of the alternatives according to the closeness coefficient, CC_i , in decreasing order. The best alternative is closest to the FPIS and farthest to the FNIS.

A worked example using the algorithm is presented in Appendix 1. An automated fuzzy tool using a high level programming language was developed based on the fuzzy TOPSIS algorithm as shown in Figure 1 below.

Automated Fuzzy Ranking Tool

Precision ranking of requirements at the touch of a button

Username:

Password:

[Rank Requirements](#)
 [Rank Security Requirements](#)
 [Rank Criteria](#)

Figure 1: Login Screen of the Automated Fuzzy Tool

Overview of the Automated Fuzzy Tool

The automated fuzzy tool was developed using the Design Science Research Methodology (DSRM) stages.

Stage 1: Identify problem and motivate

The search for a new method of requirements prioritisation was driven by the following. Firstly, requirements prioritisation is imperative in Agile Software Development. Secondly, the selection of requirements is based on multiple decision makers and multiple evaluation criteria. After conducting an extensive desktop literature review of requirements prioritisation research, a new approach using the mathematical fuzzy TOPSIS algorithm as a basis is presented. It is aligned specially for use in Agile Software Development and will ensure that the selection of

requirements meet the satisfaction of project goals and more importantly the customers' expectations.

Stage 2: Define objectives of a solution

The following objectives were defined:

Investigate current practice of security requirements prioritisation and design an automated fuzzy tool that can promote best practice in Agile Software Development; investigate how the automated fuzzy tool can be merged with conventional practices to prioritise the product/sprint backlogs and construct an automated fuzzy tool artifact for improving Multi-Criteria Decision Making (MCDM) when ranking requirements using the fuzzy TOPSIS method as a basis for the development.

Stage 3: Design & development

The design and development of the Automated Fuzzy Software Tool was implemented in an object oriented paradigm using C#.

Stage 4: Demonstration

The automated fuzzy tool was demonstrated and tested by software developers using the test website www.fuzzytool.co.za.¹ The tool was demonstrated using an illustrative example similar to appendix 1. The results were viewed by stakeholders. The demonstration was validated and verified showing the construction of all intermediate judgement matrices, distance measurement calculations and how final closeness coefficient (CCi) values were computed using illustrated examples. The stakeholders were given a walk-through of the calculations to prove the correctness of the results given by the tool at the time of the demonstration. These calculations shown manually to stakeholders took place programmatically at the backend of the automated fuzzy tool web application.

Analysis of Results

The results were analysed under stage 5 of the Design Science Research Methodology.

Stage 5: Evaluation

Software developers occupying various developer roles were interviewed to provide feedback on their evaluation of the automated fuzzy tool. The artifact was evaluated individually by three representatives of intended users of the web application tool. The evaluators represented different Agile Software Development roles namely, customer, business analyst and project manager. The evaluation was conducted at the study sites using requirements from various types of projects. For example, large projects with many requirements, small projects with few requirements, projects with detailed requirements and projects with a mix of detailed and high level requirements. The evaluation framework focused on the following themes as synthesized and referenced from the literature review namely (Hasan, Mahmood,

¹ The online web application is currently live.

Alam, Hasan, & Rahman, 2010; Tuunanen & Kuo, 2015): ease of use; total time taken; scalability; suitability to high level requirements; suitability to detailed requirements; correctness and scale of measurements.

Each theme is presented below from the data analysis of evaluators' responses.

Ease of use

This evaluation criterion describes end users experience on how easy it was to use the automated fuzzy tool. The informants indicated that the tool had simple instructions that were fairly easy to follow. Informants indicated that the use of the drop down list boxes and radio buttons ensured that user input was swift and error free. However the following were cited as potential improvements: Increase text field sizes to accommodate longer inputs and outputs; provide better hints to users during input and provide a graphical output for the end user.

Total time taken

This evaluation theme focused on the time taken to complete the process. In general evaluators indicated that the process was very quick. Time delays experienced were based on how long it took users to make a decision on a requirement.

Scalability

There were no issues reported on the ability of the tool to handle projects of various sizes.

Suitability to high level requirements

High level requirements are not specified with a great deal of detail and is elicited in the early stages of a project. The consensus reached was that the automated fuzzy tool is efficient for high level requirements.

Suitability to detailed requirements

Detailed requirements are specified generally after the analysis of requirements. Generalised requirements become more clearly defined as the project progresses. All participants indicated that the tool is suitable for detailed requirements as the prioritisation was completed just as efficiently as high level requirements.

Correctness

The correctness of the tool was illustrated to evaluators through a stepwise walkthrough of the manual calculations using their live input. The nature of the illustration was as per the demonstration mentioned in stage 4 above. Calculations were presented at the feedback session after the demonstration. All evaluators were satisfied with the explanations and accepted the results of the tool as accurate.

Scale of measurements

The automated fuzzy tool is based on fuzzy TOPSIS which presents output on a ratio scale (Karlsson, Wohlin, & Regnell, 1998). All evaluators agreed that the scale of measurement was able to present the output with the magnitude of the rating and not just output an ordered list of requirements. Evaluators found information on the magnitude extremely useful because if the differences between two requirements were marginal, more practical decisions can be taken during implementation, for

example, a requirement ranked slightly lower could be implemented immediately if resources for the higher rated requirement were not available for some reason. Magnitude also gives the developers a general idea on the importance of a requirement.

It was suggested that it would be more meaningful to the end users if the magnitude was expressed graphically. This will be most valuable as differences in CCI values of requirements can be easily detected at a glance rather than inspecting the numerical CCI values. All informants were in agreement that this modification will enhance the presentation of the output as well as make the tool easier to interpret.

Expected Value for developers

The automated fuzzy tool provides the following benefits to requirements engineers, namely, decision making is more accurate and precise (ratio data) ensuring that security requirements that are truly valuable gets implemented first. It promotes collective decision making by ensuring that a number of stakeholders must make their input before a final decision is reached. It also prevents biases and autocratic leaders from influencing decision making during security risk analysis. This was a common complaint in the field work. An additional application of the tool is its use in ranking normal requirements in the product backlog and the sprint backlogs.

Conclusion

The overall results yielded by the automated fuzzy tool were trustworthy and correct. The tool worked efficiently for a mix of high level as well as detailed requirements. The tool can be used at any stage in Agile RE, for example at the initial planning stage when requirements are more high level or prior to the iterations, after requirements analysis when more detailed requirements are specified or during the security risk analysis stage when ranking security requirements. The tool can be used by Agile RE practitioners to ensure that the security requirements prioritisation process is inclusive of all stakeholders and decisions are based on logical principles. Further research on the use of the software tool is envisaged.

References

- Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(1), 568-585.
- Hasan, M. S., Mahmood, A., Alam, M. J., Hasan, S., & Rahman, F. (2010). An evaluation of software requirement prioritization techniques. *International Journal of Computer Science and Information Security (IJCSIS)*, 8(9).
- Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14-15), 939-947.

- Martin, H. A., Zarchi, M. K., Azizollahi, S. (2011). The Application of Fuzzy Topsis Approach to Personnel Selection for PAdir Company, Iran. *Journal of Management Research* 3(2), 1-13.
- Sodhi, B., Prabhakar T.V. . (2012). A Simplified Description of Fuzzy Topsis. Kanpur: Dept. of Computer Science and Engineering.
- Souag, A., Mazo, R., Salinesi, C., & Comyn-Wattiau, I. (2015). Reusable knowledge in security requirements engineering: a systematic mapping study. *Requirements Engineering*, 1-33.
- Tuunanen, T., & Kuo, I.-T. (2015). The effect of culture on requirements: a value-based view of prioritization. *European Journal of Information Systems*, 24(3), 295-313.

APPENDIX 1

Application of Fuzzy TOPSIS to Requirements Engineering

In this section an illustrated example from a small scale project shows the application of fuzzy TOPSIS to rank security requirements. To promote understanding a step by step description is given. The mathematical translation follows each step. The example is an application on how fuzzy TOPSIS is used to show the preferences by decision makers to security requirements from a list of security requirements.

A security risk analysis has resulted in four security requirements being identified. Due to constraints in budget, only the top two requirements can be accepted into the product backlog. Prioritise these security requirements for an online book store project, consisting of the following elicited security requirements:

- A1 Sales information to be modified only by members of the sales team.
- A2 Modification to any data in the system must be logged.
- A3 Prevent SQL injections on all input fields.
- A4 Allow authorized users to access the system.

Three decision makers, namely the project manager (DM1), the book store client (DM2) and the business analyst (DM3) must rank the requirements based on the following benefit criteria:

- C1 Business value: Valuable to the business.
- C2 Size: Capable of finishing in a sprint.
- C3 Effort: Mental effort is manageable to implement in a sprint.
- C4 Core: The system must have this feature.

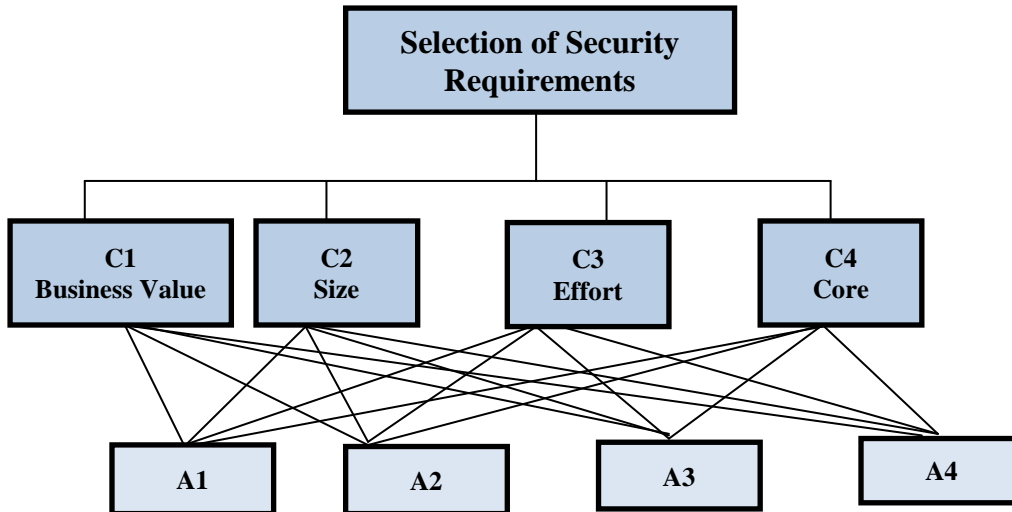


Figure 1: Hierarchical structure of selection process of the Top 2 security requirements

Figure 1 shows the hierarchical structure for the selection of the best two requirements. Each decision maker must rate each requirement on the four decision making criteria. This is a multiple criteria and multiple decision making problem. Fuzzy TOPSIS will be used to solve this multi-criteria, multi-decision making problem. The steps for the solution are given below with explanations.

Step 1: Determine the fuzzy weighting of evaluation criteria and fuzzy weighting of alternatives

(i) *Linguistic scales for evaluation criteria must be represented as a triangular fuzzy number*

Criteria

- C1 Business value
- C2 Size
- C3 Effort
- C4 Core

The fuzzy ratings for the criteria expressed as a fuzzy triangular number are illustrated in Table 1. The rating scale is determined by decision makers through consensus. In this example the linguistic rating scale ranges are from Very Low (VL) to Very High (VH) and the fuzzy rating scale ranges from 1 to 9. Any appropriate rating scale can be chosen.

Table 1: Fuzzy ratings for the criteria by decision makers

Linguistic term	Fuzzy triangular number
Very Low (VL)	(1,1,3)
Low (L)	(1,3,5)
Medium (M)	(3,5,7)
High (H)	(5,7,9)
Very High (VH)	(7,9,9)

(ii) *Represent linguistic scales for alternatives as a triangular fuzzy number. The alternatives in this instance are the set of requirements from the project backlog.*

Requirements (Alternatives)

- A1 Sales information to be modified only by members of the sales team

- A2 Modification to any data in the system must be logged
- A3 Prevent SQL injections on all input fields
- A4 Allow authorized users to access the system

Table 2: Fuzzy ratings for the alternatives by decision makers

Linguistic term	Fuzzy triangular number
Very weak (VW)	(1,1,3)
Weak (W)	(1,3,5)
Average (A)	(3,5,7)
Good (G)	(5,7,9)
Very Good (VG)	(7,9,9)

Step 2: Expert decision makers must rate the criteria and the alternatives with respect to criteria using appropriate linguistic variables. Construct aggregated fuzzy weights vector for criteria and construct the aggregated fuzzy decision matrix for alternatives.

- (i) *Decision makers DM_1 , DM_2 and DM_3 weigh the criteria using the linguistic rating scale for criteria based on their expert knowledge and experience.*

Table 3 shows decision makers ratings of criteria, for example criterion 1 (C_1) were rated as H (DM_1), H (DM_2) and L (DM_3) using linguistic scales.

Table 3: Linguistic Scales for Rating Criteria by Decision Makers

Criteria	Decision-makers		
	DM1	DM2	DM3
C1	H	H	L
C2	H	VH	M
C3	VH	VH	VL
C4	H	H	M

- (ii) *The decision makers DM_1 , DM_2 and DM_3 ratings must be represented as triangular fuzzy numbers in terms of the numeric rating scale for criteria.*

Table 4 shows the conversions from a linguistic scale to a triangular fuzzy number. For example the rating for C_1 by DM_1 is H (5, 7, 9), DM_2 is H (5, 7, 9) and DM_3 is L (1, 3, 5) respectively.

Table 4: Fuzzy weights for criteria

Criteria	Decision-maker		
	DM1	DM2	DM3
C_1	5,7,9	5,7,9	1,3,5
C_2	5,7,9	7,9,9	3,5,7
C_3	7,9,9	7,9,9	1,1,3
C_4	5,7,9	5,7,9	3,5,7

Now taking C_1 as an example, the aggregated rating for C_1 is calculated as follows (min (5+5+1), 7+7+3/3, max (9+9+5)). Similarly the aggregated rating for all other criteria are calculated. Then assemble the fuzzy decision matrix of criteria as shown in the Table 5.

Table 5: Aggregate Fuzzy Weights for criteria

Criteria	Weight		
	C1	1	5.7
C2	3	7	9
C3	1	6.3	9
C4	3	6.3	9

Mathematically aggregate weights of each criterion are calculated as follows:

Let $k =$ number of decision makers

Let $j = 1, 2, \dots, n$ representing the criteria

$$\tilde{w}_j^k = (w_{j1}, w_{j2}, w_{j3})$$

$$\text{where: } w_{j1} = \min_k \{w_{jk1}\}, w_{j2} = \frac{1}{K} \sum_{k=1}^k w_{jk2}, w_{j3} = \max_k \{w_{jk3}\} \quad (1)$$

(iii) Expert decision makers rating for the alternatives with respect to criteria using the appropriate linguistic variables. Thereafter the aggregated fuzzy decision matrix is constructed.

Table 6 shows that decision makers rating of alternatives in terms of the criteria based on linguistic the rating scale for alternatives.

Table 6: Linguistic assessments for 4 alternatives by decision makers

Criteria	Alternatives											
	A1			A2			A3			A4		
	DM1	DM2	DM3	DM1	DM2	DM3	DM1	DM2	DM3	DM1	DM2	DM3
C1	A	VG	W	A	W	A	G	VG	VW	G	A	VG
C2	VG	A	VG	W	A	VG	A	G	W	VG	VG	VW
C3	VW	W	A	G	G	W	A	VG	G	VW	A	VG
C4	VG	G	G	VG	W	G	A	VW	A	VG	G	G

Table 7 is populated by representing each entry in Table 6 as triangular fuzzy numbers in terms of the rating scale for alternatives.

Table 7: Aggregated fuzzy decision matrix

Criteria	Alternatives											
	A1			A2			A3			A4		
	DM1	DM2	DM3	DM1	DM2	DM3	DM1	DM2	DM3	DM1	DM2	DM3
C1	3,5,7	7,9,9	1,3,5	3,5,7	1,3,5	3,5,7	5,7,9	7,9,9	1,1,3	5,7,9	3,5,7	7,9,9
C2	7,9,9	3,5,7	7,9,9	1,3,5	3,5,7	7,9,9	3,5,7	5,7,9	1,3,5	7,9,9	7,9,9	1,1,3
C3	1,1,3	1,3,5	3,5,7	5,7,9	5,7,9	1,3,5	3,5,7	7,9,9	5,7,9	1,1,3	3,5,7	7,9,9
C4	7,9,9	5,7,9	5,7,9	7,9,9	1,3,5	5,7,9	3,5,7	1,1,3	3,5,7	7,9,9	5,7,9	5,7,9

Now the decision maker's fuzzy ratings are aggregated. Table 6 shows that alternative 1 based on criterion 1 was rated as A (DM₁), VG (DM₂) and W (DM₃). The decision makers corresponding fuzzy rating were A (3, 5, 7), VG (7, 9, 9) and W (1, 3, 5) as shown in Table 7 above. Therefore the aggregated rating of the 3 decision makers for alternative 1 using criterion 1 is (1, 5.7, 9) as shown in Table 8. In the same way assemble the aggregated fuzzy decision matrix for all other alternatives as shown in the Table 8.

Aggregate weights of each alternative are calculated as follows:

Let $k = \text{number of decision makers}$

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

Aggregate weightings $\tilde{x}_{ij} = (l_{ij}, m_{ij}, u_{ij})$ of alternatives (i) with respect to each criterion (j) based on the fuzzy ratings by the k^{th} decision maker is expressed by the calculations below:

$$l_{ij} = \min_k \{l_{ij}^k\}, m_{ij} = \frac{1}{K} \sum_{k=1}^k m_{ij}^k, u_{ij} = \max_k \{u_{ij}^k\} \quad (2)$$

The aggregated fuzzy decision matrix is shown in Table 8.

Table 8: Aggregated fuzzy decision matrix

Criteria	Alternatives											
	A1			A2			A3			A4		
C1	1	5.7	9	1	4.3	7	1	5.7	9	3	7	9
C2	3	7.7	9	1	5.7	9	1	5	9	1	6.3	9
C3	1	3	7	1	5.7	9	3	7	9	1	5	9
C4	5	7.7	9	1	6.3	9	1	3.7	7	1	7.7	9

The aggregate fuzzy decision matrix for the alternatives (\tilde{D}) is expressed below:

$$\tilde{D} = \begin{matrix} & C_1 & C_2 & C_j & C_m \\ \begin{matrix} A_1 \\ A_i \\ A_n \end{matrix} & \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{1j} & \tilde{x}_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \tilde{x}_{nj} & \tilde{x}_{nm} \end{bmatrix} \end{matrix} \quad (3)$$

Step 3: Normalise the fuzzy decision matrix

Normalise the fuzzy decision matrix of the alternatives (\tilde{D}). All criteria are benefit criteria. A matrix is normalized to ensure that the value for each entry is in the interval [0,1]. It ensures that the values of each criterion are comparable. In fuzzy data the normalization is achieved by dividing each entry in the fuzzy triangular number by max value of the triple for each alternative (Sodhi 2012). For example consider the entry (1, 5.7, 9) for criterion 1 for alternative1 illustrated in Table 8. Normalisation is achieved by dividing each value in the triple by 9. Therefore the normalized entry is $(1/9, 5.7/9, 9/9) = (0.333, 0.633, 1)$ as shown in Table 9.

Table 9: Normalised Aggregated fuzzy decision matrix for alternatives

	A1			A2			A3			A4		
C1	0.333	0.633	1	0.111	0.478	0.778	0.111	0.633	1	0.333	0.778	1
C2	0.333	0.856	1	0.111	0.633	1	0.111	0.556	1	0.111	0.7	1
C3	0.111	0.333	0.778	0.111	0.633	1	0.333	0.778	1	0.111	0.556	1
C4	0.556	0.856	1	0.111	0.7	1	0.111	0.411	0.778	0.111	0.856	1

The normalized fuzzy decision matrix (\tilde{R}) is written as follows:

Let $\tilde{R} = \text{normalized decision matrix}$

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

Let $(l_{ij}, m_{ij}, u_{ij}) = \text{aggregated weighting of alternative (i) with respect to criterion (j)}$

Let $\tilde{r}_{ij} = \text{triangular fuzzy number of alternative (i) with respect to criterion (j) in matrix } r \text{ with } n \text{ rows and } s \text{ columns}$

The normalized decision matrix is calculated as follows:

$$\widetilde{R} = [\tilde{r}_{ij}]_{n \times s}$$

where $\tilde{r}_{ij} = \left(\frac{l_{ij}}{u_j^+}; \frac{m_{ij}}{u_j^+}; \frac{u_{ij}}{u_j^+} \right)$ and $u_j^+ = \max_i u_{ij}$ (benefit criteria) (4)

therefore $u_j^+ = \max_i (9,9,9)$

Step 4: Compute the weighted normalised decision matrix

Table 10 shows the aggregated weights for each criteria. This table is used together with Table 11 to calculate the weighted normalized decision matrix, for example, the weighted normalized value for alternative 1 (A₁) is obtained by multiplying the criterion 1 (C₁) the values for C₁ (1, 5.7, 9) from Table 10 by the elements (0.333, 0.633, 1.00) from the normalized fuzzy decision matrix in Table 11.

Table 10: Aggregate fuzzy weights for criteria

Criteria	Weight		
C1	1	5.7	9
C2	3	7	9
C3	1	6.3	9
C4	3	6.3	9

Table 11: Normalised Aggregate fuzzy decision matrix for alternatives

	A1			A2			A3			A4		
C1	0.333	0.633	1	0.111	0.478	0.778	0.111	0.633	1	0.333	0.778	1
C2	0.333	0.856	1	0.111	0.633	1	0.111	0.556	1	0.111	0.7	1
C3	0.111	0.333	0.778	0.111	0.633	1	0.333	0.778	1	0.111	0.556	1
C4	0.556	0.856	1	0.111	0.7	1	0.111	0.411	0.778	0.111	0.856	1

Table 12 shows the completed weighted normalized fuzzy decision matrix for alternatives.

Table 12: Weighted Normalized Fuzzy Decision Matrix for Alternatives

	A1			A2			A3			A4		
C1	0.333	3.61	9	0.111	2.723	7	0.111	3.61	9	0.333	4.433	9
C2	1	5.989	9	0.333	4.433	9	0.333	3.889	9	0.333	4.9	9
C3	0.111	2.1	7	0.111	3.99	9	0.333	4.9	9	0.111	3.5	9
C4	1.667	5.39	9	0.333	4.41	9	0.333	2.59	7	0.333	5.39	9

The weighted normalized decision matrix, \widetilde{V} is computed as follows:

Let $j = 1, 2, \dots, n$ represent the criteria
 Let $i = 1, 2, \dots, s$ represent the alternatives

The weights of the evaluation criteria, (\tilde{w}_j) is multiplied by the elements \tilde{r}_{ij} of the normalized fuzzy decision matrix.

$$\widetilde{V} = [\tilde{v}_{ij}]_{n \times s}$$

Where $\tilde{v}_{ij} = \tilde{r}_{ij} \times \tilde{w}_j$ (5)

Step 5: Fuzzy Positive-Ideal solution (FPIS) values and the Fuzzy Negative-Ideal solution (FNIS) values are determined

The best performance values for each alternative are referred to as the FPIS and the worst performance values are referred to as FNIS (Vinodh et al. 2010). Table 13 shows the weighted normalized fuzzy decision matrix for alternatives. This table is used to calculate the FPIS and FNIS for alternatives under each criterion.

Table 13: Weighted Normalized Fuzzy Decision Matrix for Alternatives

	A1			A2			A3			A4		
C1	0.333	3.61	9	0.111	2.723	7	0.111	3.61	9	0.333	4.433	9
C2	1	5.989	9	0.333	4.433	9	0.333	3.889	9	0.333	4.9	9
C3	0.111	2.1	7	0.111	3.99	9	0.333	4.9	9	0.111	3.5	9
C4	1.667	5.39	9	0.333	4.41	9	0.333	2.59	7	0.333	5.39	9

Table 14 shows the FNIS and FPIS values. Table 2.16 is used for the calculation of FPIS and FNIS, for example row 1 of Table 2.16 show that the FNIS value is 0.111 and the FPIS value is 9. These values are now written in row 1 of Table 14 under FNIS(A-) and FPIS(A+) respectively.

Table 14: FNIS and FPIS

FNIS(A-)			FPIS(A+)		
0.111	0.111	0.111	9.000	9.000	9.000
0.333	0.333	0.333	9.000	9.000	9.000
0.111	0.111	0.111	9.000	9.000	9.000
0.333	0.333	0.333	9.000	9.000	9.000

The Fuzzy Positive Ideal Solution (FPIS, A⁺) and the Fuzzy Negative Ideal Solution (FNIS, A⁻) is calculated as follows:

Let $j = 1, 2, \dots, n$ represent the criteria

$$A^+ = \{ \tilde{v}_1^+, \tilde{v}_j^+, \dots, \tilde{v}_n^+ \} \quad (6)$$

$$A^- = \{ \tilde{v}_1^-, \tilde{v}_j^-, \dots, \tilde{v}_n^- \} \quad (7)$$

where $\tilde{v}_j^+ = (1, 1, 1)$ and $\tilde{v}_j^- = (0, 0, 0)$

Step 6: Calculate the distance of each alternative from FPIS and FNIS

Table 15 shows the distance values for each alternative. For example, the distance between alternative 1 criterion 1 (0.333,0.361,9.00) from Table 13 and FNIS for criterion 1 (0.111,0.111,0.111) from Table 14 is calculated as follows:

$$d = \sqrt{\frac{1}{3} [(0.333 - 0.111)^2 + (0.361 - 0.111)^2 + (9.00 - 0.111)^2]} \\ = 5.517$$

Table 15: Distances from FPIS and FNIS for alternatives

	d ⁻				d ⁺			
	A1	A2	A3	A4	A1	A2	A3	A4
C1	5.517	4.254	5.515	5.708	5.892	6.388	6.002	5.656
C2	5.987	5.535	5.408	5.656	4.935	5.656	5.809	5.535
C3	4.140	5.599	5.831	5.492	6.599	5.891	5.535	6.035
C4	5.844	5.530	4.064	5.793	4.719	5.662	6.330	5.420

The distances d_j^+ and d_j^- of each alternative respectively from \tilde{v}_j^+ and \tilde{v}_j^- are calculated as follows:

Let $j = 1, 2, \dots, n$ represent the criteria

Let $i = 1, 2, \dots, s$ represent the alternatives

$$d_i^+ = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^+) \quad (8)$$

$$d_i^- = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^-) \quad (9)$$

Where d represents the distance between two triangular fuzzy numbers. For example the distance $d(\tilde{v}_{ij}, \tilde{v}_j^+)$ where $\tilde{v}_{ij} = (l_1, m_1, u_1)$ and $\tilde{v}_j^+ = (l_2, m_2, u_2)$ is expressed as follows:

$$d(\tilde{v}_{ij}, \tilde{v}_j^+) = \sqrt{\frac{1}{3} [(l_1 - l_2)^2 + (m_1 - m_2)^2 + (u_1 - u_2)^2]} \quad (10)$$

Step 7: Obtain the closeness coefficient (CC_i)

Table 16 shows the values for the Closeness Coefficients of the four alternatives. The alternative with the **highest closeness coefficient** represents the best alternative as it is closest to the FPIS and farthest from FNIS.

Table 16: Closeness Coefficients of 4 Alternatives

	A1	A2	A3	A4
d-	21.488	20.918	20.818	22.649
d+	22.145	23.597	23.676	22.647
CC_i	0.492	0.470	0.468	0.500

The Closeness Coefficient values of alternative 1 calculated as 21.488 (**d-**) / (21.488 (**d-**) + 22.145 (**d+**)) is 0.492. Similarly the values for the closeness coefficient of other alternatives in Table 2.19 are calculated.

The closeness coefficient (CC_i) is calculated as follows:

$$CC_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (11)$$

Step 8: Define the Ranking of Alternatives

Define the ranking of the alternatives according to the closeness coefficient, CC_i, in decreasing order. The best alternative is closest to the FPIS and farthest to the FNIS.

Order:

A4 Allow authorized users to access the system

A1 Sales information to be modified only by members of the sales team

A2 Modification to any data in the system must be logged

A3 Prevent SQL injections on all input fields

The results of fuzzy TOPSIS indicate that only A4 and A1 must be sent to the product backlog.

APPENDIX 2
INTERVIEW QUESTIONNAIRE: TOOL EVALUATION

Preamble

Thank you for volunteering to participate in this research. A new software tool has been developed to address the prioritisation of clients security requirements using a computerized Fuzzy based system as a more effective and efficient way to rank normal requirements and non-functional requirements.

The purpose of this questionnaire is to therefore:

- **Elicit the views of requirements engineers on the extent to which the new software tool is able to meet their needs.**

INTERVIEW QUESTIONS: Guided questions for stakeholders of the Agile Project
SECTION A: GENERAL

1. *Researcher:* What is your role in the Agile Software Development project?
Respondent:
2. *Researcher:* What techniques are presently used to rank client requirements?
Respondent:
3. *Researcher:* What are the limitations of current techniques?
Respondent:
4. *Researcher:* Who are the stakeholders involved in the ranking of client requirements presently?
Respondent:

SECTION B: EASE OF USE

1. *Researcher:* Describe your experience with the ease of use of the software tool?
Respondent:
2. *Researcher:* What changes would you make to improve the ease of use?
Respondent:
3. *Researcher:* Comment on your satisfaction with the user interface. Does it promote easy Use?
Respondent:

SECTION C: FUNCTIONALITY AND CAPABILITY OF THE SOFTWARE TOOL

1. *Researcher:* Discuss your satisfaction with the time taken by the tool to effectively rank requirements?
Respondent:
2. *Researcher:* Comment on the correctness of the automated fuzzy tool?
Respondent:
3. *Researcher:* Comment on the scale of measurement used?
Respondent:

4. *Researcher:* Comment on the scalability of the tool. How is the performance of the tool as the number of requirements increases?

Respondent:

5. *Researcher:* Is the tool suitable for high level requirements as well as detailed requirements?

Respondent:

SECTION D: OVERALL

Researcher: Final comments on the automated fuzzy software tool in comparison to current Methods?

Respondent: