Fuzzy_MoSCoW: A Fuzzy based MoSCoW Method for the Prioritization of Software Requirements

Khadija Sania Ahmad[1], Nazia Ahmad[2], Hina Tahir[1], and Shaista Khan[1]

[1]Department of Computer Science and Engineering
Faculty of Engineering and Technology, Al-Falah University, Dhauj, Faridabad, INDIA
E-mail: serg.bsoft@gmail.com; ahmadsania18@gmail.com; kshaista3@gmail.com; hina.tahir123@gmail.com

[2]Department of Computer Science
Imam Ibdulrahman Bin Faisal University, Formerly Dammam University, Dammam, SAUDI ARABIA
E-mail: nahmad@uod.edu.sa

***Abstract*: After elicitation of software requirements, prioritization of software requirements is an important issue of requirements elicitation process. Different methods have been developed for the prioritization of the software requirements like analytic hierarchy process, TOPSIS, MoSCoW, etc. Among these methods, MoSCoW method has received less attention by software requirements engineering community that how to apply the MoSCoW method under fuzzy environment for software requirements prioritization (SRP). Therefore, to address the above issue, we proposed a method for SRPusing fuzzy based MoSCoW method. Finally, we explain the proposed method with the help of Library Management System, as a case study.**

Keywords: Software requirements elicitation, Requirements prioritization, MoSCoW method, Library Management System

I. INTRODUCTION

Requirements engineering (RE) is a process which is employed to find out the "*functional requirements*" (FR) and "*non-functional requirements*" (NFR) of software according to the need of the different types of the stakeholders [12]. Software requirements prioritization (SRP) is considered as a "*multi-criteria decision making*" (MCDM) method; and its objective is to prioritize the software requirements on the basis of different criteria's like cost, performance, usability, etc. [2]. SRP is used in requirements negotiation and release planning [1, 11]. Prioritized software requirements helps in planning software releases since all the elicited software requirements cannot be implemented due to "*limited resources*", "*inadequate budget", insufficient skilled programmers",* etc.

Achimugu *et al.* [1] presented a systematic review of 48 SRP methods. Some of these methods are given below: "Win-Win, Wiegers' Matrix Approach, Weighted Critical Analysis, Value Based Intelligent Requirement Prioritization, Value Oriented Prioritization, Value Based Requirements Prioritization, Top Ten Requirements, Theme Screening / Scoring, Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Cumulative Voting, Software Engineering Risk: Understanding and Management(SERUM), Simple Multi-Criteria Rating Technique by Swing, Round-the-Group Prioritization, Requirement Uncertainty Prioritization Approach(RUPA), Requirements Triage, Relative Weighting, Rank Based on Product Definition, Ranking, Quality Functional Development (QFD), Priority Groups, Planning Game, Ping Pong Balls, Pairwise Comparison, Attributed Goal Oriented Requirements Analysis (AGORA) method, Numerical Assignment Technique,MoSCoW", etc. [1, 3, 4].

In [1], authors have shown that MoSCoW is the most cited and utilized method of the SRP. Therefore, in this paper, we mainly focus on the MoSCoW method. MoSCoW technique prioritizes the requirements on the basis of the following criteria: (i) "*Must Have*" (Mo), (ii) "*Should Have*" (S), (iii) "*Could Have*" (Co), and (iv) "*Won't Have This Time*" (W). The meaning of these criteria's is given below:
(i) *Must have (Mo)* defines the requirements that must be included in the final software product, (ii) In *Should have(S)*requirements,high priority requirement should be included if possible, within the delivery time frame, (iii) *Could have (Co)* requirement are desirable or nice to have requirements and could be included without incurring too much effort or cost, and (iv) *Won't have this time (W)* requirement are those requirements that stakeholder want to have but all the stakeholders have the consensus that requirements will not be implemented in the current version of the time.

Based on our literature review, we identify that SRP techniques like, AHP, TOPSIS, and AGORAmethods, etc., have been developed under fuzzy environment to deal with the imprecise and unambiguous information during decision making process. For example, Chamodrakas and Martakos [5] apply "*utility based fuzzy TOPSIS method for selection of efficient network in heterogeneous wireless network*". Sadiq and Jain [12] presented an extended version of AGORA method under fuzzy environment. AGORA method was also extended by Mohammad *et al.* [10] in 2016; and it is known as "*Fuzzy Attributed Software Goal Oriented Requirements Analysis*" (FAGOSRA) method. Therefore, it motivates us to extend the MoSCoW method under fuzzy environment. The remaining part of this paper is organized as follows: Related work in the area of SRP is given section II. Fuzzy based MoSCoW for SRP is presented in section III. In section IV, we explain the proposed method by considering the case study, which is based on Library Management System (LMS). Finally, we conclude our research work in section V; and we also discuss the future work in the same section, i.e., section V.

II. RELATED WORK

Svensson *et al.* [15] considered 11 software companies for the prioritization of the software requirements. In their work they found that quality requirements (QR) have lower priority than the functional requirements (FR). Sadiq and Jain [13] proposed a fuzzy oriented method for the prioritization of the software requirements. They have proposed the fuzzy based method to strengthen the "*goal oriented requirements elicitation processes*"(GOREP). Angelis *et al.* [3]proposed a multivariate "*compositional data analysis*" framework for the analysis of the data obtained from the "*Cumulative Voting*" method.

Baber *et al.* [4] identified that most of the SRP methods are suitable only for the small set of the software requirements. On the basis of literature review, it has been observed that SRP methods have the following limitations: "(i) most of the SRP methods don't provide a scalable solutions when the numbers of requirements are scaled up and the results are generally faulty and error prone (ii) most of the methods only deal with the small set of the software requirements". Therefore, in [4] a fuzzy based method was proposed for SRP with the help of neural networks.

Perini *et al.* [11] proposed a Case-Based Ranking (CBRank) using machine learning based approach for SRP. In their work, authors compare the results of the CBRank with Analytic Hierarchy Process (AHP). In 2015, a controlled experiment was setup to prioritize the software requirements by McZara *et al.* [9]. In their work, authors proposed a method for SRP, which is based on natural language processing and satisfiability modulo theories solvers; and they call it "SNIPR". SNIPR method was applied by the 40 software engineers for the selection of 20 requirements from the set of 100 requirements. As a result authors show that SNIPR method consumes less time and it is more accurate than the weighted sum model.Based on our review, we identify that in literature, little attention is given for the prioritization of software requirements under fuzzy environment. Therefore, an attempt has been made in this paper to prioritize the software requirements using fuzzy based MoSCoW method.

III. FUZZY BASED MOSCOW APPROACH

In this section, we present the fuzzy based method for the prioritization of software requirements using MoSCoW. Proposed method has the following steps:

- **Step 1:**Identify the functional and non –functional requirements of the software
- **Step 2:**Define triangular fuzzy numbers (TFN) for the Mo, S, Co, and W
- **Step 3:**Collect the decision maker's fuzzy assessment
- **Step 4:**Apply graded mean integration representation on TFN
- **Step 5:**Compute the ranking values of the requirements

Step 1: Identify the functional and non –functional requirements of the software

In this step, we identify the functional and non-functional requirements of software. Functional requirements (FR) are used to "*describe the functionality of the software*"; and Non-functional requirements (NFR) are employed to describe "*how the system is supposed to be*". For the identification of the requirements of the software, we used "*goal oriented requirements elicitation process*" (GOREP). In GOREP, requirements analyst decomposes and refines the goal of an organization or enterprise into sub-goals. The process of decomposition and refinement continues until the responsibility of the last goal is assigned to some agents or some systems. During decomposition and refinement, two types of the connectives are present among the sub-goals, i.e. AND connectives and OR connectives. Resultant structure of this process looks like an AND/OR graph. This graph is used to identify the FR and NFR of the system [2, 12, 13].

Step 2: Define triangular fuzzy numbers (TFN) for the Mo, S, Co, and W

In real life application, stakeholders specify their preferences using linguistic variables, for example, "*Kerala is one of the most populartourist destination of India*", and "*Delhi has good number of reputed Central Universities for studies*". In these examples, "*most popular*"and the "*good*"are the linguistic terminologies (LT). In order to deal these LT, in 1965, Lotfi Zadeh introduced the concept of the Fuzzy Logic (FL) [16 17]. In FL, different types of the fuzzy numbers are used like "triangular fuzzy numbers (TFN)", "trapezoidal fuzzy numbers", "pentagonal fuzzy numbers", etc. In our work, we apply TFN because of its simplicity in representation and computation during decision making process [12, 13].The objective of this step is to define the TFN for the Mo ("*Must have*"), S ("*Should have*"), Co ("*Could have*"), and W ("*Won't have*").

Step 3: Collect the decision maker's fuzzy assessment

As we know that different decision makers are involved during the software prioritization process. Therefore, the objective of this step is to collect the decision maker's fuzzy assessment for the given set of FR and NFR.

Step 4: Apply graded mean integration representation on TFN

To find out the ranking values of the software requirements, in this step we use the "*canonical representation of multiplication operation*" associated with L^{-1} R^{-1} inverse arithmetic principle and the "*graded mean integration representation*" on TFN proposed by Chen and Hsieh in 1998 [6]. Several researchers have applied the method proposed by Chen and Hsieh [6]for the ranking of alternative and requirements in real life applications [7, 8, 13]. For example, Sadiq and Neha [14] applied the method proposed by [6] for the "*elicitation of testing requirements from the selected set of FR*".

Let $W_1 = (m_1, q_1, s_1)$ and $W_2 = (m_2, q_2, s_2)$ be two TFNs as shown in Fig. 1.

In Fig. 1, $L_W(x) = \frac{x-m}{q-m},\ m \leq x \leq q$, and $R_W(x) = \frac{x-s}{q-s},\ q \leq x \leq s$;

$$L_W^{-1}(h) = m + (q-m)h \qquad 0 \leq h \leq 1$$
$$R_W^{-1}(h) = s + (q-s)h \qquad 0 \leq h \leq 1$$

Here, $L_W(x)$ and $R_W(x)$ are the function L and R of the fuzzy number W, respectively. $L_W^{-1}(x)$ and $R_W^{-1}(x)$ are the inverse functions of $L_W(x)$ and $R_W(x)$ at level h.

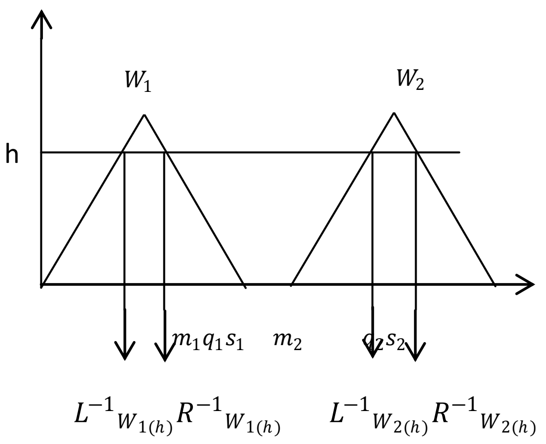

Fig. 1: Representation of two TFNs

The multiplication of W_1 and W_2 at h-level can be computed as [6]:

$$W_1(h) \times W_2(h) = (L^{-1}{}_{W_{1(h)}} L^{-1}{}_{W_{2(h)}}, L^{-1}{}_{W_{1(h)}} R^{-1}{}_{W_{2(h)}} \; R^{-1}{}_{W_{1(h)}} L^{-1}{}_{W_{2(h)}}, R^{-1}{}_{W_{1(h)}} R^{-1}{}_{W_{2(h)}}) \tag{1}$$

The graded mean integration representation of W_1 and W_2 at h-level is given as below:

$$P\,(W_1 \times W_2) = \int_0^1 \int_0^1 \int_0^1 \Big[\left(h_{W_1} L^{-1}{}_{W_{1(h)}}\right)\left(h_{W_2} L^{-1}{}_{W_{2(h)}}\right) + \left(h_{W_1} L^{-1}{}_{W_{1(h)}}\right)\left(h_{W_2} R^{-1}{}_{W_{2(h)}}\right) + \left(h_{W_1} R^{-1}{}_{W_{1(h)}}\right)\left(h_{W_2} L^{-1}{}_{W_{2(h)}}\right) + \left(h_{W_1} R^{-1}{}_{W_{1(h)}}\right)\left(h_{W_2} R^{-1}{}_{W_{2(h)}}\right) \Big] \times \frac{h_{W_1 W_2} dh_{W_1} dh_{W_2} dh_{W_1 W_2}}{\int_0^1 h_{W_1} dh_{W_1} \int_0^1 h_{W_2} dh_{W_2} \int_0^1 h_{W_1 W_2} dh_{W_1 W_2}} \tag{2}$$

After simplification, we have the following results:

$$P\,(W_1 \times W_2) = \left(\frac{m_1 + 4q_1 + s_1}{6}\right) \times \left(\frac{m_2 + 4q_2 + s_2}{6}\right) \tag{3}$$

Step 5: Compute the ranking values of the requirements

In this step we apply the equation (3) to compute the ranking values of the software requirements. Requirements which have the highest ranking values would be implemented during the first release of the software. Requirements which have the lowest ranking values would be implemented during the last release of the software.

IV. CASE STUDY

In this section, we apply the proposed method to prioritize the requirements of Library Management System (LMS).The objective of the LMS is to issue the books to the students of various courses like Engineering, Medical Sciences, Natural Sciences, Architecture and Ekistics, etc. The system will show the availability of the books to the student; and will record the priority of the student depending on the submission of its need in the system. If the book has already been issued by the librarian to the students, then after submitting the books by students, that book would be issued to those students who have the highest priority. The information about the priority would be informed to the student by SMS on his/her mobile. If the student will not return the book on the specified date, there would be fine of some rupees, say 20 RS per day.

Step 1: To identify the FR and NFR of LMS, we apply the GOREP. After applying the GOREP, we have identified the following FR and NFRs:

FR1: Login module of LMS
FR2: Check the availability of the books
FR3: To generate the priority of the students
FR4: Information of availability of books through SMS
FR5: Candidates who have submitted books after due date
FR6: Download the research papers from Journals
FR7: Download the research papers from Conference Proceedings
FR8: Late book fee submission
FR9: Inform the Chief Librarian, if the book is not available for a particular course
FR10: Report to Chief Librarian, if the book has lost

NFR1: The system should be secure
NFR2: The system should not be costly
NFR3: The system should be reliable

Step 2: Once we have identified the FR and NFR, then the next step is to define triangular fuzzy numbers (TFN) for the Mo, S, Co, and W. Here, Mo, S, Co, and W are the linguistic variables (LV).
The LV for the importance weight of each FR is given in Table I. The LV for the evaluation of NFR is given in Table II.

Table I: LV for FR

LV	TFN
Won't have (W)	(0, 0, 0.33)
Could have (Co)	(0, 0.33, 0.66)
Should have (S)	(0.33, 0.66, 1)
Must have (Mo)	(0.66, 1, 1)

Table II: LV for the evaluation of NFR

LV	TFN
Could have (CoH)	(2.5, 2.5, 5)
Should have (SoH)	(2.5, 5, 7.5)
Must have (MoH)	(5, 7.5, 1)
Won't have (WH)	(7.5, 1,1)

Step 3: Now we collect the decision maker's (DM) fuzzy assessment. Here, we assume that only one (DM) is participating in requirements prioritization process. Evaluation of each FR by DM with respect to NFR is given in Table III.

Table III: Evaluation of FR with respect to NFR

NFGs	FG	LV
NFR1	FR1	Mo
	FR2	Co
	FR3	S
	FR4	Mo
	FR5	So
	FR6	Mo
	FR7	Mo
	FR8	S
	FR9	S
	FR10	C
NFR2	FR1	S
	FR2	C
	Fr3	S
	FR4	S
	FR5	Mo
	FR6	S
	FR7	S
	FR8	Mo
	FR9	Mo
	FR10	Mo
NFR3	FR1	Mo
	FR2	S
	Fr3	S
	FR4	S
	FR5	S
	FR6	C
	FR7	C
	FR8	Mo
	FR9	S
	FR10	S

In our case study, we have identified ten FR and three NFRs. The importance weight of each NFR is given in Table IV.

Table IV: The importance weight of each NFR

NFR	Importance weight
NFR1	MoH
NFR2	CoH
NFR3	SH

Step 4: Now we apply the equation 3 to compute the ranking values of the FR. In this step, we first calculate the ranking value of the FR1. The detailed calculation of FR1 is shown below:

$$(0.66, 1, 1) \otimes (5, 7.5, 1) + (0.33, 0.66, 1) \otimes (2.5, 2.5, 5) + (0.66, 1, 1) \otimes (2.5, 5, 7.5)$$
$$FR1 = 12.29651$$

Similarly, we have calculated the values of the remaining FRs. The values of the remaining FRs are given below

$$FR2 = 6.2453$$
$$FR3 = 9.19451$$
$$FR4 = 10.88651$$
$$FR5 = 10.015$$
$$FR6 = 9.23151$$
$$FR7 = 9.23151$$
$$FR\ 8 = 11.42513$$
$$FR9 = 10.015$$
$$FR\ 10 = 8.02913$$

Step 5: On the basis the above ranking values, we identify that FR1 has the highest ranking value (RV), i.e. 12.29651 and FR2 has the lowest ranking value, i.e., 6.2453.

Suppose the client want to implement top five software requirements during the first release of the software. So according to the above ranking values, following requirements would be implemented:

FR1: Login module of LMS*(RV = 12.29651)*
FR8: Late book fee submission*(RV = 11.42513)*
FR4: Information of availability of books through SMS*(RV = 10.88651)*
FR5: Candidates who have submitted books after due date*(RV = 10.015)*
FR9: Inform the Chief Librarian, if the book is not available for a particular course*(RV = 10.015)*

V. CONCLUSIONS AND FUTURE WORK

In this paper, we present a fuzzy based MoSCoW method for the prioritization of the software requirements, i.e., *Fuzzy MoSCoW*. There are five steps in the proposed method, i.e., (i)Identify the functional and non –functional requirements of the software, (ii) define triangular fuzzy numbers (TFN) for the Mo, S, Co, and W, (iii) collect the decision maker's fuzzy assessment, (iv) apply graded mean integration representation on TFN, (iv) compute the ranking values of the requirements. We have applied proposed method for the prioritization of the requirements of Library Management System (LMS). In the proposed method for the elicitation of the software requirements, we have applied "*goal oriented requirements elicitation process*" (GOREP). After applying GOREP, we have identified ten FR and three NFR for LMS. Finally, we have selected following five requirements that would be implemented during the first release of the software, i.e., FR1, FR8, FR4, FR5, and FR9. Future work includes the following:

- To prioritize the FR of the LMS when multiple stakeholders participate during requirements prioritization process under fuzzy environment

- To apply the proposed for the prioritization which have the large number of requirements, for example, 50 requirements, 100 requirements, and/ or 1000 requirements

Acknowledgement

This work has been carried out in collaboration with the Software Engineering Research Group of *Bakewarr Software Solutions*(BSS), New Delhi-110025, India.

References

1. Achimugu P., Selamat A., Ibrahim R., and Mahrin M. N., "A Systematic Literature Review of Software Requirements Prioritization Research", Information and Software Technology, Vol. 56, pp. 568-585, 2014.
2. Afreen N., Khatoon A., and Sadiq M. 2015. A Taxonomy of Software's Non-Functional Requirements. In: Proceedings of the Second International Conference on Computer and Communication Technologies, Vol. 379 of Advances in Intelligent Systems and Computing, Springer, pp. 47-53.
3. Angelis P. C. L., Rovegard P., and Wohlin C., "Prioritization of Issues and Requirements by Cumulative Voting: A Compositional Data Analysis Framework", EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 361-370, 2013.
4. Baber M. I., Ramzan M., and Ghayyur S. A. K., "Challenges and Future Trends in Software Requirements Prioritization", IEEE International Conference on Computer Networks and Information Technology, 2011.
5. Chamodrakas I and Martakos D., "A utility-based fuzzy TOPSIS method for energy efficient network selection in heterogeneous wireless network", Applied Soft Computing, Vol. 11, pp. 3734-3743, 2011.
6. Chen S. H. and Hsieh C. H., "Graded Mean Integration Representation of Generalized Fuzzy Numbers" proceedings of the 6th International Conference on Fuzzy Theory and Its Applications, Taiwan, pp. 1-6, 1998.
7. Chou C-C. 2003,"The Canonical Representation of Multiplication Operation on Triangular Fuzzy Numbers", Computers and Mathematics with Applications, Vol. 45, pp. 1601-1610, 2003.
8. Chou C-C., "The Representations of Multiplication Operation on Fuzzy Numbers and Application to Solving Fuzzy Multiple Criteria Decision Making Problems", PRICAI, LNAI, Springer, pp. 161-169., 2006.
9. McZara J., Sarkani S., Holzer T, and Eveteigh T, "Software Requirements Selection and Prioritization based on Natural Language Processing and Satisfiability Modulo Theories Solvers-A Controlled Experiment", Empirical Software Engineering, Springer, Vol. 20, pp. 1721-1761, 2015.
10. Mohammad C. W., Shahid M., Husain S. Z., "FAGOSRA: Fuzzy Attributed Goal Oriented Software Requirements Analysis Method", 9th IEEE International Conference on Contemporary Computing, pp.384-389, 2016.
11. Perini A., Susi A, and Avesani P., "A Machine Learning Based Approach to Software Requirements Prioritization", IEEE Transactions on Software Engineering, Vol. 39, No. 4, 2013.
12. Sadiq and Jain, "A Fuzzy Based Approach for Requirements Prioritization in Goal Oriented Requirements Elicitation Process", 25th International Conference on Software Engineering and Knowledge Engineering, USA, 2013.
13. Sadiq M. and Jain S K. 2015. A Fuzzy Based Approach for the Selection of Goals in Goal Oriented Requirements Elicitation Process. International Journal of System Assurance Engineering and Management, Springer, Vol. 6, Issue 2, pp. 157-164.
14. Sadiq M. and Neha, "Elicitation of Testing Requirements from the Selected Set of Software's Functional Requirements using Fuzzy Based Approach", International Conference on Computational Intelligence and Data Mining, Springer, December 10-11, 2016, Organized by KIIT University, Bhubaneswar, Odisha, India.
15. Svensson R. B., Gorschek T., Regnell B., Torkar R., Shahrokni A., Feldt R., and Aurum, "Prioritization of Quality Requirements: State of Practice in Eleven Companies", 19th IEEE International Requirements Engineering Conference, pp. 69-78, 2011.
16. Zadeh L. A. 1965. Fuzzy Sets. Information Control, Vol. 8, pp. 338-353.
17. Zadeh L. A. 1996. Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems, Vol. 4, Issue 2, pp. 103-111.