IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-9, Sept. 2018 E-ISSN: 2347-2693

Software Requirements Selection Using Consistent Pairwise Comparison Matrices of AHP

M. Sadiq[1*], S. Khan[2], C. W. Mohammad[2]

[1]Software Engineering Research Group, Computer Engineering Section, UPFET Jamia Millia Islamia (A Central University), New Delhi, 110025, INDIA

[2]Computer Science and Technology Research Group, Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), New Delhi-110025, INDIA

**Corresponding Author:* sadiq.jmi@gmail.com, +91-9891667600

Available online at: www.ijcseonline.org

Accepted: 12/Sept/2018, Published: 30/Sept/2018

Abstract— Analytic hierarchy process (AHP) is one of the important multi-criteria decision making algorithms which is used to rank the software requirements on the basis of different criteria like performance, usability, reliability, cost, etc. In the area of software engineering, different methods have been developed to rank the software requirements using AHP like PRGFOREP, GOASREP, etc. Based on our literature review we identify that in software requirements selection (SRS) less attention is given to check the consistency of the "*pairwise comparison matrices*" (PCM). The ranking values of the software requirements would be consistent only when the PCM would be consistent. Therefore, to address this issue we proposed a method for SRS by generating the different patterns and sub-patterns of the PCM. In our case study, we have generated the 8 patterns and for each pattern we have generated the 64 sub-patterns. As a result, we have generated 512 sub-patterns of PCM and stored the results into a database so that the information stored in the database could be used for requirements analysis. The applicability of the proposed method is explained with the help of a case study.

Keywords—Software requirements selection, AHP, Pairwise comparison matrices, Types of requirements

I. Introduction

Software requirements engineering (RE) is a process which is used to develop the successful software product. We divide the RE into five sub-processes, i.e., "requirements elicitation", "requirements modeling", "requirements analysis", "requirements verification and validation", and "requirements management" [1]. Among these sub-process, requirements elicitation is the key process to identify the need of the different types of the stakeholders. In this process, different stakeholders are involved during group elicitation process so that software requirements can be identified. One of the difficult activities of the requirements elicitation process is the selection of software requirements (SR). "Software requirements selection" (SRS) on the basis of different criteria creates a "multi-criteria decision making" (MCDM) problem; and its aim is to choose "those requirements that would be implemented during different releases of the software". In software engineering, "non-functional requirements" (NFR) are employed as criteria to choose the "functional requirements" (FR) from the list of the FRs [2]. Different types of the MCDM techniques have been used in SRS like "analytic hierarchy process" (AHP) [3], "techniques for order preference by similarity to ideal solutions" (TOPSIS), etc. [4]. Karlsson et al. [5] investigated the following methods for the prioritization of SR, i.e., "AHP", "hierarchy AHP", "spanning tree", "bubble sort", "binary search tree" and "priority groups". As a result they identify that AHP is one of the useful and trustworthy methods for the software requirements (SR) prioritization in industrial applications. Therefore, it motivates us to apply the AHP for the SRS.

Different methods have been developed to select the SR. For example, Sadiq and Jain [6] developed a method for the "selection of software goals in goal oriented requirements elicitation process" [2]. Garg et al. [7] proposed "GOASREP: Goal Oriented Approach for Software Requirements Elicitation and Prioritization using Analytic Hierarchy Process". In AHP, "pairwise comparison matrices" (PCM) are employed to compare different alternatives based on different criteria. In literature, we identify that AHP has been applied for SRS but checking the consistency of the PCM has received less attention by the researchers and academicians [8]. There are some studies, which have focused on the generation of the consistent results on the basis of the consistent PCM. In [8] Sadiq and Afrin generated the different patterns and sub-patterns for the PCM of size 3X3, by considering three requirements. In their study, they have generated 8 patterns and for one pattern they have generated 48 sub-patterns. After generating the 48 different patterns, a database was maintained in which the values of the entire consistent ratio were stored. Whenever the decision maker will generate any PCM, the pattern of that PCM will be compared with the results of the database. If the database suggests that PCM is valid then only the PCM would be used during the SRS process, otherwise, the database will suggest you for other PCM. This work was extended by Khan et al. [9] in which 64 patterns was generated for 8 patterns; and as a result they have generated 512 patterns for the PCM of size 3X3. Their work was limited to only "non-functional requirements" (NFRs). Therefore, the objective of this paper is to extend our previous work [9] and select "functional requirements" on the basis of NFRs when the PCM are consistent.

The remaining part of this paper is structured as follows: In section II we discuss the related work in the area of SRS which are based on MCDM algorithms. A brief introduction about AHP is given in section III. Proposed method for SRS when PCM are consistent is given in section IV. A case study based on Institute Examination System is given in Section V. Section VI contains the conclusion and future work.

II. RELATED WORK

This section presents related work in the area of SRS. Different methods or techniques have been developed for the selection of SR. For example, Karlsson and Ryan [10] proposed a method to determine the set of requirements of project under limited resources. In their work authors have applied AHP to elicit the requirements contribution. Requirements are pair-wise compared on the basis of the importance and cost of the requirements. Ruhe et al. [11] proposed a method "for trade off-analysis for the selection of software requirements". In their work, AHP was also used to elicit the preferences of the stakeholders related to the various classes of the requirements. In 2009, Sadiq et al. [12] proposed an approach for the elicitation of the SR in which AHP has been used to prioritize the SR. This work was extended by the same research group in which AHP was applied for the prioritization of the SR [13]. Apart from this, there are some research areas in which AHP has been used. For example, "selection of software development life cycle models" using AHP [14]. Lai et al. [15] conducted a case study based on one of the popular MCDM algorithm, i.e., AHP; and developed a "group decision making" (GDM) system for the selection of multi-media authorizing system. As a result authors find out that AHP is an effective method to develop the consensus among the stakeholders in GDM system. Wei et al. [16] proposed an essential structure for the selection of ERP to support the following: "business goal and strategies of an enterprise", "identify the appropriate attributes", etc. Min [17] proposed a method for selecting a proper logistics software using AHP to deal with "qualitative and quantitative factors in MCDM environments". Schniederjans and Wilson [18] proposed an information system selection method using AHP with goal programming model framework. Karsak and Ozogul [19] proposed a "framework for ERP software selection".

Apart from the above studies, we have identified some other studies in which SRS/goal selection have been used as one of the important steps of the different methodologies used in goal oriented requirements engineering [6], stakeholder identification methods [1], etc. For example, Sadiq et al. [20] apply the AHP in "goal oriented requirements elicitation method for the prioritization of the software requirements" to develop the AHP-GORE_PSR methodology. This methodology was extended by Sadiq and Afrin [8] in which they have generated the "different patterns of PCM to check whether the PCM are consistent or not". In recent studies, Sadiq and Nazneen [21] proposed a method for the "elicitation of software testing requirements from the selected set of software's functional requirements in goal oriented requirements elicitation process". In another study, a method for stakeholder identification was proposed by Sadiq [1] on the basis of the importance of SR. Importance of the SR was determined by the "selected set of the software requirements". On the basis of our literature review, we find out that in the above studies, AHP has been widely used in SRS but without considering whether the PCM is consistent or not. Therefore, in our study we mainly focus on the consistency of the PCM during SRS.

III. ANALYTIC HIERARCHY PROCESS

AHP is a popular MCDM algorithm which was developed by the "Thomas L. Saaty in 1980" for the selection as well as prioritization of the alternatives on the basis of the different criteria [3]. In [24] Saaty presented the principles and philosophy of the theory used in AHP. AHP has been widely used in software engineering. For the evaluation of the alternatives on the basis of different criteria's like security [22] authenticity [23], cost [1], etc., Saaty [24] proposed a scale to specify the preferences of one alternative over another. One common scale, adopted by the Saaty is shown in Table 1. In AHP, PCM are used to specify the preferences of the stakeholders. Different algorithms have been developed to compute the priority values of the PCM. In our work, following algorithm is used to compute the "ranking values" (RV) of the PCM [3, 24]:

TABLE 1: Saaty Rating Scale

Intensity of importance	Definition
1	"Equal Importance"
3	"Somewhat more important"
5	"Much more important"
7	"Very much more important"
9	"Absolutely more important"
2,4,6,8	"Intermediate values"

Algorithm to compute the ranking values of SR:

Step 1: Add the column of the PCM and store the result in $Column_{PCM}$

Step 2: Normalized the $Column_{PCM}$ and store the results in Normalized_ $Column_{PCM}$

Step 3: Take the average of the row from Normalized_ $Column_{PCM}$. As a result the we will get the priorities of the alternatives and store the results in P_1, P_2, …P_N. Where N is the total number of requirements

Step 4: Multiply the first column with P_1, second column with P_2, and N^{th} column with P_N; and store the results in Weighted_Column

Step 5: Calculate the sum of each row from Weighted_Column; and store the results in Weighted_Sum (WS) as WS_1, WS_2, …WS_N

Step 6: Divide the elements of the WS_1, WS_2, …WS_N by the P_1, P_2, …P_N as:

$$\lambda_1 = WS_1/ P_1, \lambda_2 = WS_2/ P_2 \ldots \lambda_N = WS_N/ P_N$$

Step 7: Compute the average of λ_1, λ_2 and λ_N; and store the results in λ_{max}

Step 8: Calculate the "*consistency ratio*" (CR) by using the following equation:

$$CR = \frac{\frac{\lambda max - N}{N-1}}{RI}$$

Fig. 1: Algorithm to compute the ranking values of SR

Here, RI is the "*consistency index of a randomly generated*" PCM. The value of the RI for 3, 4, 5, 6 requirements would be 0.58, 0.9, 1.12, and 1.24, respectively [3, 24].

IV. PROPOSED METHOD

The steps of SRS using consistent PCM are given below*:*

Step 1: Apply goal oriented method to elicit the FR and NFRs

Step 2: Construct PCM and generate different patterns and sub-patterns of PCM.

Step 3: Compute the CR of each PCM

Step 4: Identify those patterns in which the value of the CR is less than 10%

Step 5: Select the software requirements on the basis of the consistent PCM.

Explanations of the above steps are given in next section, i.e., Case Study.

V. CASE STUDY

In our work, we apply the proposed method to select the SR of "*Institute Examination System*" (IES). The explanation of the proposed method in the context of IES is given below:

Step 1: Apply goal oriented method to elicit the FR and NFRs

There are different methods to elicit the SR like traditional methods, group elicitation method, cognitive method, and contextual method. These methods focuses on WHAT of a proposed system rather than WHY. Goal oriented method (GOM) focuses on WHY and it visualizes the different types of the SR like FR and NFR using AND/OR graph. In GOM, "*high level objective of an organization are refined and decomposed until the responsibility of the goals/sub-goals are assigned to some agents and systems"*. After applying the GOM, we have identified the following FR and NFR of IES:

List of FRs:

"FR1: Login module of IES for different types of the users, i.e., students, teacher, and the administration

FR2: To send the SMS/email on the mobile of the students or parents to submit the examination fee before appearing in the examination

FR3: To display the results of the students of different courses

FR4: System should generate the seating arrangement and the same should be forwarded on the mobile/email of the students

FR5: To generate the date sheet of the theory and practical courses

FR6: To generate the hall ticket of the eligible students

FR7: Filling of semester/annual examination form

FR8: Approve examination form

FR9: Online conduct of examination for those courses which requires multiple objective based question papers

FR10: To enter the internal assessment marks and the end semester marks of the theory and the practical courses"

List of NFRs:

NFR1: Security; **NFR2:** Performance; **NFR3:** Usability

Step 2: Construct PCM and generate different patterns and sub-patterns of PCM.

To start the ranking process using AHP, first PCMs are constructed after evaluating the FRs based on the NFRs. In our study, we have ten FR and three NFR. For ten FR there would be 1024 patterns and in the case of 3 NFR, there would be eight different patterns. In our case study, for three NFRs following patterns have been generated:

Pattern 1: When NFR1 is favourable over NFR2, when NFR1 is favourable over NFR3, and when NFR2 is favourable over NFR3

Pattern 2: When NFR1 is favourable over NFR2, when NFR1 is favourable over NFR3, and when NFR3 is favourable over NFR2

Pattern 3: When NFR1 is favourable over NFR2, when NFR3 is favourable over NFR1, and when NFR2 is favourable over NFR3

Pattern 4: When NFR1 is favourable over NFR2, when NFR3 is favourable over NFR1, and when NFR3 is favourable over NFR2

Pattern 5: When NFR2 is favourable over NFR1, when NFR1 is favourable over NFR3, and when NFR2 is favourable over NFR3

Pattern 6: When NFR2 is favourable over NFR1, when NFR1 is favourable over NFR3, and when NFR3 is favourable over NFR2

Pattern 7: When NFR2 is favourable over NFR1, when NFR3 is favourable over NFR1, and when NFR2 is favourable over NFR3

Pattern 8: When NFR2 is favourable over NFR1, when NFR3 is favourable over NFR1, and when NFR3 is favourable over NFR2.

For the explanation point of view, we generate the first sub-pattern for pattern 1 in Fig. 2.

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	3
NFR2	1/3	1	3
NFR3	1/3	1/3	1

Fig. 2: First sub-pattern for pattern 1, i.e., when NFR1 is favourable over NFR2, when NFR1 is favourable over NFR3, and when NFR2 is favourable over NFR3

Now we apply the algorithm, as shown in Fig. 1, to calculate the "*ranking values*" (RV) of NFRs. For sub-patterns 1, as shown in Fig. 2, we find out that the value of the CR is 0.1524. Similarly, we compute the consistency ratio of all the sub-patterns.

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	3
NFR2	1/3	1	3
NFR3	1/3	1/3	1

Fig. 3: Sub-pattern 1.1

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	3
NFR2	1/3	1	5
NFR3	1/3	1/5	1

Fig. 4:Sub-pattern 1.2

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	3
NFR2	1/3	1	7
NFR3	1/3	1/7	1

Fig. 5:Sub-pattern 1.3

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	3
NFR2	1/3	1	9
NFR3	1/3	1/9	1

Fig. 6:Sub-pattern 1.4

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	5
NFR2	1/3	1	3
NFR3	1/5	1/3	1

Fig. 7:Sub-pattern 1.5

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	5
NFR2	1/3	1	5
NFR3	1/5	1/5	1

Fig. 8:Sub-pattern 1.6

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	5
NFR2	1/3	1	7
NFR3	1/5	1/7	1

Fig. 9:Sub-pattern 1.7

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	5
NFR2	1/3	1	9
NFR3	1/5	1/9	1

Fig. 10:Sub-pattern 1.8

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	7
NFR2	1/3	1	3
NFR3	1/7	1/3	1

Fig. 11:Sub-pattern 1.9

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	7
NFR2	1/3	1	5
NFR3	1/7	1/5	1

Fig. 12:Sub-pattern 1.10

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	7
NFR2	1/3	1	7
NFR3	1/7	1/7	1

Fig. 13:Sub-pattern 1.11

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	7
NFR2	1/3	1	9
NFR3	1/7	1/9	1

Fig. 14:Sub-pattern 1.12

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	9
NFR2	1/3	1	3
NFR3	1/9	1/3	1

Fig. 15:Sub-pattern 1.13

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	9
NFR2	1/3	1	5
NFR3	1/9	1/5	1

Fig. 16:Sub-pattern 1.14

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	9
NFR2	1/3	1	7
NFR3	1/9	1/7	1

Fig. 17:Sub-pattern 1.15

NFRs	NFR1	NFR2	NFR3
NFR1	1	3	9
NFR2	1/3	1	9
NFR3	1/9	1/9	1

Fig. 18:Sub-pattern 1.16

NFRs	NFR1	NFR2	NFR3
NFR1	1	5	3
NFR2	1/5	1	3
NFR3	1/3	1/3	1

Fig. 19:Sub-pattern 1.17

NFRs	NFR1	NFR2	NFR3
NFR1	1	5	3
NFR2	1/5	1	5
NFR3	1/3	1/5	1

Fig. 20:Sub-pattern 1.18

NFRs	NFR1	NFR2	NFR3
NFR1	1	5	3
NFR2	1/5	1	7
NFR3	1/3	1/7	1

Fig. 21:Sub-pattern 1.19

Similarly, we generate the remaining PCM. Then we compute the CR of each PCM in next step.

Step 3: Compute the CR of each PCM

In this step, we compute the "*consistency ratio*" (CR) of all the sub-patterns of pattern 1; and the results for pattern are exhibited in Table II.

Table II: Database of CR of the sub-patterns 1 to 64

Sub-patterns	CR
Sub-pattern 1.1	0.1524
Sub-pattern 1.2	0.3298
Sub-pattern 1.3	0.4901
Sub-pattern 1.4	0.6380
Sub-pattern 1.5	**0.0477**
Sub-pattern 1.6	0.1646
Sub-pattern 1.7	0.2802
Sub-pattern 1.8	0.3886
Sub-pattern 1.9	**0.0092**
Sub-pattern 1.10	**0.0833**
Sub-pattern 1.11	0.1711
Sub-pattern 1.12	0.2570
Sub-pattern 1.13	**0.0000**
Sub-pattern 1.14	**0.0390**
Sub-pattern 1.15	0.1053
Sub-pattern 1.16	0.1715
Sub-pattern 1.17	0.5113
Sub-pattern 1.18	0.6458
Sub-pattern 1.19	0.8688
Sub-pattern 1.20	1.0600
Sub-pattern 1.21	0.1891
Sub-pattern 1.22	0.4010
Sub-pattern 1.23	0.5812
Sub-pattern 1.24	0.7377
Sub-pattern 1.25	**0.0960**
Sub-pattern 1.26	0.2633
Sub-pattern 1.27	0.4154
Sub-pattern 1.28	0.5502
Sub-pattern 1.29	**0.0449**
Sub-pattern 1.30	0.1757
Sub-pattern 1.31	0.3056
Sub-pattern 1.32	0.4242
Sub-pattern 1.33	0.6066
Sub-pattern 1.34	0.9635
Sub-pattern 1.35	1.2408
Sub-pattern 1.36	1.4737
Sub-pattern 1.37	0.3559
Sub-pattern 1.38	0.6502
Sub-pattern 1.39	0.8867
Sub-pattern 1.40	1.0849
Sub-pattern 1.41	0.2181
Sub-pattern 1.42	0.4644
Sub-pattern 1.43	0.6723
Sub-pattern 1.44	0.8494
Sub-pattern 1.45	0.1338
Sub-pattern 1.46	0.3406
Sub-pattern 1.47	0.5250
Sub-pattern 1.48	0.6856
Sub-pattern 1.49	0.8453
Sub-pattern 1.50	1.2818
Sub-pattern 1.51	1.6102
Sub-pattern 1.52	1.8792
Sub-pattern 1.53	0.5326
Sub-pattern 1.54	0.9043
Sub-pattern 1.55	1.1939
Sub-pattern 1.56	1.4312
Sub-pattern 1.57	0.3540
Sub-pattern 1.58	0.6736
Sub-pattern 1.59	0.9339
Sub-pattern 1.60	1.1508
Sub-pattern 1.61	0.2401
Sub-pattern 1.62	0.5163
Sub-pattern 1.63	0.7514
Sub-pattern 1.64	0.9516

Step 4: Identify those patterns in which the value of the CR is less than 10%

In the case of the first pattern, we observe that the value of CR is less that 10% for the following sub-patterns of pattern 1, i.e., sub-pattern 1.5, 1.9, 1.10, 1.13, 1.14, 1.25, and 1.29. Similarly, we identify the sub-patterns of all the patterns in which the value of the CR is less than 10%, see Table II.

Step 5: Select the software requirements on the basis of the consistent PCM

In this step, we select the FR from the set of ten FR on the basis of three NFR only when the PCM is/are consistent. Now the decision makers will generate the PCM for the NFRs. Suppose the decision makers generate the following PCM for the NFRs.

Non-Functional Requirements	NFR1	NFR2	NFR3
NFR1	1	3	5
NFR2	1/3	1	3
NFR3	1/5	1/3	1

Fig. 22: PCM generated by the decision maker

Now the pattern of the NFR, as shown in Fig. 22, will be compared with the patterns and sub-patterns for three NFRs. As a result, we identify that the above PCM matches with the first pattern; and within the first pattern given PCM matches with the pattern 1.5, as shown in Fig. 7. As we know that, the CR of pattern 1.5 is less than 10%. Therefore, the above PCM would be used in the selection of FR. After applying the Algorithm, as shown in Fig. 1, the ranking values of the NFRs are computed; and the final results of the NFR value are given below:

NFR1 = 0.636986, NFR2 = 0.258285, and NFR3 =0.104729.

On the basis of our analysis, we find out that NFR1, i.e., Security, has the highest priority; therefore, it would be used during the selection of the software requirements process. Now, the FR would be evaluated on the basis of the security requirements. For the ten FR, we first construct the consistent PCM of FR. The contents of the PCM for ten FR are given in Table III. The CR of the PCM, as given in Table III, is consistent; and the value of the CR is 0.0139, which is less that 10%. The database as shown in Table II would be used in our study.

Table III: Evaluation of FR on the Basis of Security Requirements

FR	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10
FR1	1	1/2	1/2	1	2	5	3	7	5	2
FR2	2	1	1	2	3	7	5	9	7	3
FR3	2	1	1	2	3	7	5	9	7	3
FR4	1	1/2	1/2	1	2	5	3	7	5	2
FR5	1/2	1/3	1/3	1/2	1	3	2	5	3	1
FR6	1/5	1/7	1/7	1/5	1/3	1	1/3	2	1	1/3
FR7	1/3	1/5	1/5	1/3	1/2	3	1	3	3	1/2
FR8	1/7	1/9	1/9	1/7	1/5	1/2	1/3	1	1/2	1/5
FR9	1/5	1/7	1/7	1/5	1/3	1	1/3	2	1	1/3
FR10	1/2	1/3	1/3	1/2	1	3	2	5	3	1

After applying the Algorithm, as shown in Fig. 1, we get the following "ranking values" (RV) of the FRs: "*FR1= 0.135, FR2, 0.223, FR3=0.223, FR4=0.135, FR5=0.079, FR6= 0.028, FR7=0.053, FR8=0.018, FR9=0.028, FR10=0.079*".

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a method for the selection of software requirements when the PCM are consistent. Proposed method includes the following steps: "*(i) identify the different types of the SR using goal oriented method, (ii) generate different patterns and sub-patterns of PCM, (iii) compute the consistency ratio of each PCM, (iv) identify those patterns which do not produce the consistent results, (v) select the SR on the basis of the consistent PCM*". RV identified by the algorithm would be used by the decision makers to select those SR that would be developed during different release of the software. If the decision makers decide that only the top three requirements would be implemented. Then FR2 and FR3 have the first priority; FR1 and FR4 have the second priority; FR5 and FR10 have the third priority. In the first release of the software following set of the requirements would be designed and developed: FR2, FR3, FR1, FR4, FR5, and FR10. Proposed method has been applied to select the SR of IES. Future work includes the following: (i) We will extend our work by evaluating the set of FR on the basis of the other two NFRs, i.e., performance and usability [25, 26]; and (ii) To perform literature review of SRS methods based on search based software engineering concepts [27-33].

REFERENCES

[1]. M. Sadiq, "A Fuzzy-Set Based Approach for the Prioritization of Stakeholders on the basis of the Importance of Software Requirements", IETE Journal of Research, Taylor and Francis, Vol. 63, pp. 616-629, 2017.

[2]. M. Sadiq, S.K. Jain, "Applying Fuzzy Preference Relation for Requirements Prioritization in Goal Oriented Requirements

Elicitation Process", International Journal of Systems Assurance Engineering and Management, Springer, Vol. 5. Pp. 711-723, 2014.
[3]. E. Mu, M. Pereyra.-Rojas, "Understanding the Analytic Hierarchy Process", Practical Decision Making, Springer, pp. 7-22, 2017
[4]. M. Behzadian, , S.K. Otaghsara, M. Yazdani, , J. Ignatius, " A State-of the –art Survey of TOPSIS Applications", Expert Systems with Applications, Elsevier, Vol. 39, pp. 13051- 13069, 2012
[5]. J. Karlsson, C. Wohlin, and B. Regnell, "An Evaluation of Methods for Prioritizing Software Requirements", Information and Software Technology, Vol. 39, pp. 939-947, 1998.
[6]. M. Sadiq, S.K. Jain, "A Fuzzy Based Approach for the Selection of Goals in Goal Oriented Requirements Elicitation Process", International Journal of Systems Assurance Engineering and Management, Springer, Vol. 6., pp.157-164, 2015.
[7]. N. Garg, M. Sadiq, and P. Agarwal, "GOASREP: Goal Oriented Approach for Software Requirements Elicitation and Prioritization using Analytic Hierarchy Process", International Conference on Frontiers in Intelligent Computing Theory and Applications, AISC, Springer, 2016.
[8]. M. Sadiq, A. Afrin, "Extending AHP-GORE-PSR by Generating Different Patterns of Pairwise Comparison Matrix", International Conference on Information, Communication and Computing Technology, CCIS, Springer-Verlag, Singapore, 2017.
[9]. S. Khan, C. W. Mohammad, and M. Sadiq, "Generating Patterns and Sub-Patterns of Pairwise Comparison Matrices for the Selection of Software Requirements", IEEE International Conference on Advances in Computing, Communication, Control, and Networking, 2018.
[10]. J. Karlsson and K. Ryan, "Supporting the Selection of Software Requirements," In the Proceedings of the IEEE 8th International Workshop on Software Specification and Design, pp. 146-149, 1996.
[11]. G. Ruhe, A. Eberlein, and D. Pfahl, "Trade-off Analysis for Requirements Selection," International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 4, pp. 345-366, 2003.
[12]. M. Sadiq, S. Ghafir, M. Shahid, "An Approach for Eliciting Software Requirements and its Prioritization using Analytic Hierarchy Process", In the Proceedings of the IEEE International Conference on Advances in Recent Technologies in Communication and Computing, pp. 790-795, 2009.
[13]. M. Sadiq, J. Ahmad, M. Asim, A. Qureshi , and S. Rani, " More on Elicitation of Software Requirements and Prioritization using AHP", In the Proceedings of the IEEE International Conference on Data Storage and Data Engineering pp. 232-236, 2010.
[14]. M. A. Khan, A. Parveen, and M. Sadiq, "A Method for the Selection of Software Development Life Cycle Models Using Analytic Hierarchy Process", In the Proceedings of the IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, pp. 539-545, 2014.
[15]. V. S. Lai, Bo. K. Wong, W. Cheung, "Group Decision Making in a Multi-criteria Environment: A Case using the AHP in Software Selection", European Journal of Operational Research, Vol. 137, No. 1, pp. 134-144, 2002.
[16]. C. C. Wei, C-F. Chien, M. J. Wang, "An AHP-based Approach to ERP System Selection", International Journal of Production Economics, Vol. 96, Issue 1, pp. 47-62, 2005.
[17]. H. Min, "Selection of Software: The Analytic Hierarchy Process", International Journal of Physical Distribution and Logistics Management, Vol. 22, No. 1, pp. 42-52, 1992.
[18]. M. J. Schniederjans, R. K. Wilson, "Using the Analytic Hierarchy Process and Goal Programming for Information System Project Selection", Information and Management, Vol. 20, No. 5, pp. 333-342, 1991.
[19]. E. E. Karsak and C. O. Ozogul, "An Integrated Decision Making Approach for ERP System Selection", Expert Systems with Applications, Vol. 36, No. 1, pp. 660-667, 2009.
[20]. M. Sadiq, T. Hassan, S. Nazneen, "AHP_GORE_PSR: Applying Analytic Hierarchy Process in Goal Oriented Requirements Elicitation Method for the Prioritization of Software Requirements", In the Proceedings of the 3rd IEEE International Conference on Computational Intelligence and Communication Technology, pp. 10-11, 2017.
[21]. M. Sadiq, S. Nazneen, "Elicitation of Software Testing Requirements from the Selected Set of Software's Requirements in GOREP", International Journal of Computational Systems Engineering, Inderscience, 2018 (in press).
[22]. S. Kathuria, "A Survey on Security Provided by Multi-clouds in Cloud Computing", International Journal of Scientific Research in Network Security and Communication, Vol. 6, Issue. 1, 2018
[23]. O. Ayankemi ONI, "A Framework for Verifying the Authenticity of Banknote on the Automated Teller Machine (ATM) using Possibilistic C-Means Algorithm", International Journal of Scientific Research in Computer Science and Engineering, Vol. 6, Issue. 2, pp. 57-63, 2018.
[24]. T. L. Saaty, "How to Make a Decision: The Analytic Hierarchy Process", European Journal of Operational Research, Vol. 48, No. 1, pp. 9-26, 1990.
[25]. M. Sadiq, F. Firoze, "A Fuzzy based Approach for the Selection of Software Testing Automation Framework", In the Proceedings of the International Conference on Computational Intelligence in Data Mining (ICCIDM), Springer, Vol. 33, pp. 335-343, 2014.
[26]. M. Sadiq, S.K. Jain, "An Insight into Requirements Engineering Processes", In the Proceedings of the 3rd International Conference on Advances in Communication, Network, and Computing Lecture Notes in Computer Science, Springer, Chennai India, pp. 313-318, 2012.
[27]. Y. Zhang, M. Harman, S.A. Mansouri, "The Multi-Objective Next Release Problem", In the Proceedings of the ACM 9th Annual Conference on Genetic and Evolutionary Computation, New York, 2007.
[28]. A. J. Bagnall, V. J. Rayward–Smith, I. M. Whittley,, "The Next Release Problem", Information and Software Technology, Vol. 43, No. 14, pp. 883-890, 2001.
[29]. J. Sagrado, I. M. Aguila, "Ant Colony Optimization for Requirements Selection in Incremental Software Development" In the Proceedings of the 1st IEEE International Symposium on Search Based Software Engineering. Cumberland Lodge, Windsor, UK, (2009)
[30]. Sagrado, J., Aguila, I.M., F. J. Orellana, F.J.: Ant Colony Optimization for the Next Release Problem: A comparative Study, In the Proceedings of the 2nd IEEE International Symposium on Search Based Software Engineering, 2010.
[31]. H. Jiang, J. Zhang, J. Xuan, Z. Ren, Y. Hu, "A hybrid ACO Algorithm for the Next Release Problem", In the Proceedings of the 2nd International Conference on Software Engineering and Data Mining, 2010.
[32]. J.Y. Souza, C.L.B. Maia, T. N. Ferreira, R.A.F. Carmo, M. M.A. Brasil, "An Ant Colony Optimization Approach to the Software release planning with Dependent Requirements", In the Proceedings of the Search Based Software Engineering, Lecture Notes in Computer Science, Springer, Vol. 6959, pp. 142-157, 2011.
[33]. J.M. Chaves-Gonzalez, M. A. Perez- Toledano, A. Navasa, "Software Requirements Optimization using Multi-objective Swarm Intelligence Evolutionary Algorithm", Knowledge Based System, Elsevier, Vol. 83,pp. 105-115, 2015

Authors Profile

Dr. Mohd. Sadiq has been working as Assistant Professor, Computer Engineering, UPFET, Jamia Millia Islamia (A Central University), New Delhi-110025, India, since 2005. He received Ph. D. degree in Computer Engineering from Department of Computer Engineering, National Institute of Technology (NIT), Kurukshetra, India, in 2017. Dr. Sadiq has worked as Post-doc Fellow at the Center for Soft Computing Research, Indian Statistical Institute, Kolkata, West Bengal, India. He has also worked as Visiting Scientist at Indian Statistical Institute, Bangalore Centre, Karnataka, India. He did M. Tech. in Computer Science and Engineering from Aligarh Muslim University, Aligarh, India, in 2005; and B.E. in Computer Science and Engineering from Al-Falah School of Engineering and Technology, Dhauj, Faridabad, affiliated with Maharshi Dayanand University, Rohtak, Haryana, in 2001, with first position in second year. Dr. Sadiq has published more than 50 research papers in International Conferences and Journals like IETE Journal of Research, Taylor and Francis; CSI Transactions on ICT, Springer; International Journal of System Assurance Engineering and Management, Springer, etc. He has reviewed several papers of International Journals like IEEE Transactions on Engineering Management, International Journal of System Assurance Engineering and Management, Springer; Journal of Computer and Electrical Engineering, Elsevier; Information and Software Technology, Elsevier; Journal of System and Software, Elsevier; Measurement, Elsevier, etc.; and conferences like IEEE- INDICON-2015, India, ICCIDM-Springer-2014, IEEE- ICACCI-2013, etc.

Mr. Shadab Khan is pursuing Ph.D. in Computer Science and Technology from Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University) New Delhi-25, under the guidance of *Prof. Chaudhary Wali Mohammad* and *Dr. Mohd. Sadiq*. He received M.Tech. in Computer Science in 2012 from M.B. University, Solan, India; and B.E. in Computer Science and Engineering from Al-Falah School of Engineering and Technology, Dhauj, Faridabad, Haryana, affiliated to Maharshi Dayanand University, Rohtak, Haryana in 2004. He has worked as Assistant Professor in Department of Computer Science and Engineering at Sunder Deep Institute of Engineering and Technology, Ghaziabad, U. P, affiliated with AKTU, Lucknow, U.P. India. His area of interest includes Software Engineering, Data Structure and Algorithms. He is the Life Member of The Indian Society for Technical Education (ISTE).

Prof. Chaudhary Wali Mohammad received B. Sc. (Hons.), M. Sc., M. Phil., and Ph. D., all degrees in Mathematics, from Department of Mathematics, Faculty of Natural Science, Aligarh Muslim University, Aligarh, U.P., India, in the year, 1978, 1980, 1983, and 1986, respectively. After completion of Ph.D. in Mathematics, Prof. Chaudhary joined Jamia Millia Islamia (JMI), New Delhi, on October 1, 1986, as a Lecturer (Contract) in Mathematics and become permanent faculty as a Lecturer (Mathematics) on February 9, 1989, in Faculty of Engineering and Technology, JMI, New Delhi. Prof. Chaudhary promoted as a Reader (Mathematics) in 1999 and become full Professor on February 9, 2007. He worked as Head of the Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, JMI, New Delhi since December 5, 2014 to December 4, 2017. Prof. Chaudhary has supervised three Ph.D. Thesis in the area of Multiple Hypergeometric Functions. His area of interest includes Multiple Hypergeometric Functions, Data Structure and Algorithms, Graph Theory, Multi-Criteria Decision Making (MCDM) algorithms and its application to Software Engineering. After teaching different papers of B. Tech/M. Tech courses, i.e., Numerical Analysis and Computer Programming (NACP), Numerical and Scientific Computing, Data Structure and Computer Programming (DCSP), Prof. Chaudhary gets motivated to work in the area of Computer Science and Technology. Currently, he is supervising four Ph.D. Thesis in the area of Computer Science and Technology; and one in the area of Applied Mathematics.

