

Maintainability assessment for software by using a hybrid fuzzy multi-criteria analysis approach

Shivani Kundu^{a*} and Kirti Tyagi^b

^aAjay Kumar Garg Engineering College, Ghaziabad, Uttar Pradesh, India

^bAssociate professor, INHA University in Tashkent

CHRONICLE

Article history:

Received: October 1, 2016

Received in revised format: November 16, 2016

Accepted: March 23, 2017

Available online:

March 23, 2017

Keywords:

Maintainability

Maintainability factors

AHP

Multi-criteria fuzzy AHP

Similarity based approach

ABSTRACT

Maintainability plays a fundamental role for achieving success in software system and it is considered as an important quality characteristics. Maintainability may be predicted efficiently by using soft computing techniques as they provide good results. In this paper similarity- based approach is used with the contribution of fuzzy Analytical Hierarchical Process (AHP) and fuzzy technique for order preference by similarity to ideal solution (TOPSIS) at 2-level hierarchy. Here similarity- based approach illustrates the combine approach of fuzzy AHP and fuzzy TOPSIS. This approach is used to provide the rank of software to select the best one for maintainability estimation. Also, several factors are presented that influence the software maintainability. These factors are taken as criterion and three software products are taken as alternatives.

© 2017 Growing Science Ltd. All rights reserved.

1. Introduction

Maintainability is the procedure of finding and correcting mistakes or bugs within instance period and under specified resource (Gyimothy et al., 2005). It may also raise efficiency of software by adding several new features. In the spirited global markets, maintainability is becoming a substance of growing importance for many companies. Planning for maintainability may minimize changes during intend and decrease maintenance-related transform orders and rework in structure. It is observed that generally maintenance activity is based on some factors which belong to software: Analyzability, Modifiability, Stability, Testability and Understandability. Maintainability may also be measured as an internal feature if it is solely derived from software system. Several models have used maintainability including its factors as one of the major characteristics of software quality. The factors influencing maintainability are Analyzability, Modifiability, Stability, Testability and Understandability.

* Corresponding author.

E-mail address: shivanikundusk@gmail.com (S. Kundu)

Multi-Criteria Analysis (MA) requires the decision maker (DM) to provide qualitative assessments. In addition, it provides many ways of disaggregating difficult problems, of measuring the scope to which options attain objectives, of weighting the objectives, and of reassembling the sections. Fortunately, many computer programs that are simple to use have been developed to support the technical features of MA. It establishes the performance of every alternative with respect to every criterion and the relative weight of the evaluation criteria with respect to the overall objective of the problem. As a result, uncertain, inaccurate and subjective data are usually present which make the decision-making process complex and challenging. The AHP of Saaty (1980) is a popular method for tackling MA problems involving qualitative data, and has successfully been applied to several actual decision states. An eigenvector method is used to solve the reciprocal matrix for determining the criteria and alternative performance. Most popular MA method is AHP, which was developed by Saaty in 1980, where alternatives and criteria are represent in hierarchical structure (Zhang, 2010).

2. Literature review

Various researchers have worked in this field to estimate the software maintainability. But maintainability estimation is much difficult for real time problems. Here some important soft computing techniques for maintainability estimation are analyzed in this section. Ghosh et al. (2012) proposed a method to compare the maintainability of object-oriented software system. The inputs for the method are size, complexity, coupling and inheritance which affect the software maintainability in different object-oriented programming languages. These inputs were determined on the basis of survey from different experts which consist of project managers, system developers, researchers and others who are working on this field. This method shows any program build in the object-oriented language C++ has highest degree of maintainability as compared to other languages.

Goel et al. (2012) developed a methodology that facilitates the qualitative evaluation of object-oriented software systems based on fuzzy approach. Maintainability model as illustrated in ISO/IEC-9126 was taken as the support model. Object-oriented factors that carried out in recognizing plan and code characteristics in object-oriented software which in turn helps in assessing software maintenance effort had also been incorporated. The results indicate that improved evaluation accuracy has been achieved by applying this model. It also reveals the effectiveness of fuzzy layered approach in predicting object-oriented software maintainability; it can be a useful and practical addition to the framework of software maintainability assessment.

Dubey et al. (2012) proposed a fuzzy model for maintainability estimation of object-oriented software system. The inputs are taken as coupling, inheritance, complexity, class, and number of children on which maintainability depends. Rule base were produced by expert's knowledge, with 243 rules for evaluating object-oriented software system. The planned model predicted the maintainability on two software systems. The results are validated by the AHP technique.

A fuzzy catalog system utilizing AHP was developed to resolve the problem of software maintainability factors by (Chen & Liu, 2009). A new synthetic maintainability evaluation model based on fuzzy entropy theory was recommended by Cai et al. (2012). They analyzed the maintainability related design factors from regular perspective and planned a fuzzy set based approach to quantitatively assess design for the maintainability.

Jeet and Dhir (2012) planned a system using fuzzy inference approach which uses activity-based quality model to contract with software maintainability. A lot more similar studies can be found elsewhere in literature. Momeni and Zahedian (2014) presented that ANFIS can more correctly predict maintainability as compared to other models. For this they chosen four metrics and used them for training, validation and testing.

Chidamber and Kemerer (1991) defined some initial suggestion for language-independent OO design factors and proposed six object oriented design factors to forecast the maintainability of OO software systems. This suite is further enhanced in 1994 and the factors suite was tested on systems developed in Smalltalk(TM) and C++ by Chidamber and Kemerer (1994). Two more factors were added into presented C&K factors suite by (Li, 1998).

The projected metrics group became quite accepted and evaluated analytically in a number of studies by different researchers which were conducted to authenticate C&K metrics suite. Many estimation models were constructed using statistical algorithms also by machine learning algorithms.

3. Hierarchical structure of software maintainability factors

Many researchers have used and developed a number of factors to forecast the maintainability of software. Here the various types of factors which influence the software maintainability as recommended by Kumar et al. (2012). So, let us take a short outline of most significant contributory factors affecting the maintainability. There are five basic factors Analyzability, Modifiability, Stability, Testability and Understandability that affects the maintainability of software (Table 1). These factors are affected by their sub factors as shown in Fig. 1.

Table 1
Maintainability relation with factors

Factors	Description	Relation	Relation Description
Analyzability (A_n)	It describes the effort required for find out the defects; fault or causes of failures or, for analyze the parts to be customized	$A_n \propto M$	Raising the level of analyzability of the component increase the maintainability of the component. So make the components as analyzable as possible.
Modifiability (M_f)	Modifiability describes the ability to alteration or modifications of the software programs.	$M_f \propto M$	Adding more detail to the component will affect and motivate the maintainability in term of increase.
Stability (S_t)	Stability refers to the risk of unforeseen cause of modifications.	$S_t \propto M$	Stability helps in understanding and organizes the module. If stability will increase then maintainability will increase.
Testability (T_e)	Testability defines the ability to authorize the alteration.	$T_e \propto M$	It is an evaluation of how simple it is to make tests for the system and its mechanisms, and to perform these tests in order to verify if the criteria are met by ISO/IEC (1996). So that maintainability is directly influence by understandability.
Understandability (U_n)	Understandability is a factor of maintainability that relate to the user's attempt for identifying the logical theory and its applicability.	$U_n \propto M$	If a module is free from errors then it will be accurate and easy to understand. So that if understandability will increase then understandability will increase.

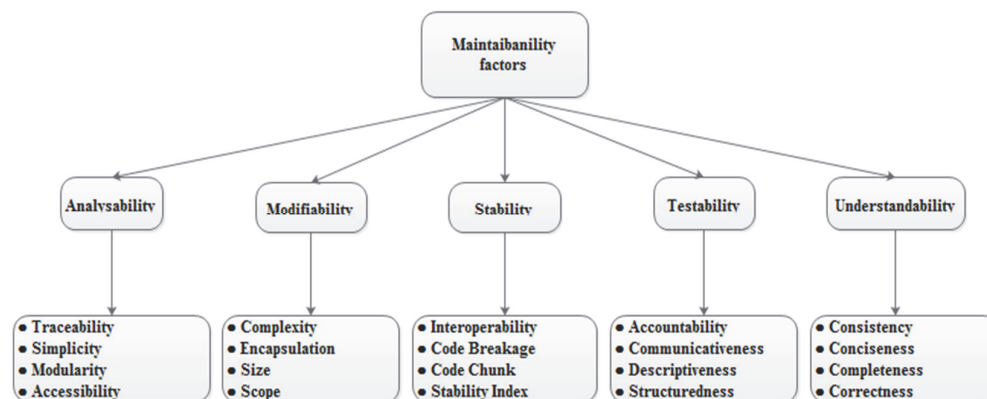


Fig. 1. Classification of Maintainability Factors

4. Material and methods

Various researchers have effort in this area to predict the software maintainability. But maintainability is more difficult for real time problem. MCDM is an approach which is associated for solving, planning, structuring and decision problems. The combination of multi-criteria analysis and pairwise comparison is used with similarity based approach.

4.1 Triangular fuzzy numbers

A fuzzy number may be symbolized as a function which is known as membership function. This membership function is signified between 0 and 1. The triangular fuzzy number (Gani & Assarudeen, 2012) is determined by triplet as (n_1, n_2, n_3) . The membership function $\mu_A(x)$ (Deng, 1999, 2007) is show in Fig. 2 and describes as:

$$\mu_A(x) = \begin{cases} \frac{(x - n_1)}{(n_2 - n_1)} & n_1 \leq x \leq n_2 \\ \frac{(n_3 - x)}{(n_3 - n_2)} & n_2 \leq x \leq n_3 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where n_1 , n_2 and n_3 are the lower bound, most possible value, and upper bound respectively (denoted by l , m , u).

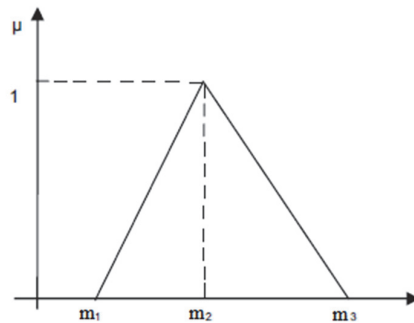


Fig. 2. Triangular membership function

For making quantitative assessment in pair-wise comparison, fuzzy numbers are used as shown in Table 2.

Table 2

Fuzzy numbers used for making qualitative assessments

Fuzzy number	Membership function
1	(1, 1, 3)
X	$(x-2, x, x+2)$ for $x = 3, 5, 7$
9	(7, 9, 11)

4.2 The MA Approach

Mahmood (1982), and Seetha Lakshmi (2013) proposed the multi criteria method where many numbers of criteria and alternatives are offered to make the decision. Most popular MCDA approach is AHP, developed by Saaty (1980). In this paper software maintainability estimation is achieved by using MA. It is a combination of fuzzy pair-wise comparison and similarity based approach. Here software product as alternatives and identified factors are used as criteria.

4.3 Similarity-based approach

Deng (2007), Moradi and Ebrahimi (2014) introduced the similarity based approach for ranking process. In MCDA, TOPSIS approach is also used to denote the multi-criteria analysis decision problem. Performance index is considered for each alternative across all criteria by using TOPSIS (Hwang & Yoon, 1981). In similarity based approach, the overall performance index is calculated by using degree of similarity between each alternative and best solution via alternative gradient and magnitude.

4.4 The basic steps of Fuzzy Multi-criteria Analysis

Step 1 By using Table 1 define the fuzzy judgment matrix either by criteria importance (W) or alternatives with respect to criteria (C_j).

$$C_j \text{ or } W = \begin{bmatrix} \bar{m}_{11} & \bar{m}_{12} & \cdots & \bar{m}_{1k} \\ \bar{m}_{21} & \bar{m}_{22} & \cdots & \bar{m}_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ \bar{m}_{k1} & \bar{m}_{k2} & \cdots & \bar{m}_{kk} \end{bmatrix} \quad (2)$$

where

$$\bar{m}_{ls} = \begin{cases} \bar{1}, \bar{3}, \bar{5}, \bar{9}, & l < s \\ 1, & l = s, l, s = 1, 2, \dots, k; \quad k = m \text{ or } n \\ \frac{1}{\bar{m}_{sl}}, & l > s \end{cases} \quad (3)$$

Step 2 The decision matrix (D) and weight vector (W) for MA can be calculated as:

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1x} \\ d_{21} & d_{22} & \cdots & d_{2x} \\ \cdots & \cdots & \cdots & \cdots \\ d_{y1} & d_{y2} & \cdots & d_{yx} \end{bmatrix} \quad (4)$$

$$W = (w_1, w_2, \dots, w_x), \quad (5)$$

where

$$d_{ij} \text{ or } w_{ij} = \frac{\sum_{s=1}^k \bar{m}_{ls}}{\sum_{l=1}^k \sum_{s=1}^k \bar{m}_{ls}}, \quad (6)$$

where $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ and $k = m$ or n , and d_{ij} and w_j are the resulting fuzzy performance of A_i ($i = 1, 2, \dots, n$) with criterion C_j .

Step 3 Fuzzy performance matrix Z will be obtained with the decision matrix (D) by multiplying the weight vector.

$$D = \begin{bmatrix} w_1 d_{11} & w_2 d_{12} & \cdots & w_x d_{1x} \\ w_1 d_{21} & w_2 d_{22} & \cdots & w_x d_{2x} \\ \cdots & \cdots & \cdots & \cdots \\ w_1 d_{y1} & w_2 d_{y2} & \cdots & w_x d_{yx} \end{bmatrix} \quad (7)$$

Step 4 Find interval performance matrix, by determining a-cut on the performance matrix, where $0 \leq a \leq 1$.

$$Z_a = \begin{bmatrix} [z_{11l}^a, z_{11r}^a] & [z_{12l}^a, z_{12r}^a] & \cdots & [z_{1xl}^a, z_{1xr}^a] \\ [z_{21l}^a, z_{21r}^a] & [z_{22l}^a, z_{22r}^a] & \cdots & [z_{2xl}^a, z_{2xr}^a] \\ \cdots & \cdots & \cdots & \cdots \\ [z_{y1l}^a, z_{y1r}^a] & [z_{y2l}^a, z_{y2r}^a] & \cdots & [z_{yxl}^a, z_{yxr}^a] \end{bmatrix}, \tag{8}$$

where

$$\begin{aligned} a_l &= (a \times (m - l)) + l, \\ a_r &= u - (a * (u - m)), \end{aligned} \tag{9}$$

Step 5 The crisp performance matrix is calculated as following matrix-

$$Z_a^{\lambda'} = \begin{bmatrix} z_{11a}^{\lambda'} & z_{12a}^{\lambda'} & \cdots & z_{1xa}^{\lambda'} \\ z_{21a}^{\lambda'} & z_{22a}^{\lambda'} & \cdots & z_{2xa}^{\lambda'} \\ \cdots & \cdots & \cdots & \cdots \\ z_{y1a}^{\lambda'} & z_{y2a}^{\lambda'} & \cdots & z_{yxa}^{\lambda'} \end{bmatrix}, \tag{10}$$

where

$$0 \leq \lambda \leq 1, \lambda z_{ijr}^a + (1 - \lambda) z_{ijl}^a$$

Since this k is a regular value and its value can be considered as $k = 0$, $k = 0.5$ and $k = 1$. These values are used to denote as pessimistic, moderate, or optimistic view, correspondingly for decision maker. So that, for calculating the crisp performance matrix, $k = 0.5$ will be using.

Step 6 Normalization process is applied on crisp performance to find the normalized performance Z_a^λ as shown below.

$$Z_a^\lambda = \begin{bmatrix} z_{11a}^\lambda & z_{12a}^\lambda & \cdots & z_{1xa}^\lambda \\ z_{21a}^\lambda & z_{22a}^\lambda & \cdots & z_{2xa}^\lambda \\ \cdots & \cdots & \cdots & \cdots \\ z_{y1a}^\lambda & z_{y2a}^\lambda & \cdots & z_{yxa}^\lambda \end{bmatrix}, \tag{11}$$

Where

$$z_{ija}^\lambda = \frac{z_{ija}^{\lambda'}}{\sqrt{\sum_{i=1}^y (z_{ija}^{\lambda'})^2}}, \tag{12}$$

Step 7 Positive ideal solution $A_a^{\lambda+}$ and negative ideal solution $A_a^{\lambda-}$ is calculated as shown below.

$$\begin{aligned} A_a^{\lambda+} &= (z_{1a}^{\lambda+}, z_{2a}^{\lambda+}, z_{2a}^{\lambda+}, \dots, z_{xa}^{\lambda+}), \\ A_a^{\lambda-} &= (z_{1a}^{\lambda-}, z_{2a}^{\lambda-}, z_{2a}^{\lambda-}, \dots, z_{xa}^{\lambda-}), \end{aligned} \tag{13}$$

where

$$\begin{aligned}
 z_{ja}^{\lambda+} &= \max(z_{1ja}^{\lambda}, z_{2ja}^{\lambda}, z_{3ja}^{\lambda}, \dots, z_{xja}^{\lambda}), \\
 z_{ja}^{\lambda-} &= \min(z_{1ja}^{\lambda}, z_{2ja}^{\lambda}, z_{3ja}^{\lambda}, \dots, z_{xja}^{\lambda}),
 \end{aligned}
 \tag{14}$$

Step 8 The similarity between each alternatives A_i and positive ideal solution and negative ideal solution is calculated as shown in Eqs. (15-16).

$$S_i^+ = \frac{(\sum_{k=1}^x z_{ik}^{\lambda+})^{\frac{1}{2}} \cos \theta_{i+}}{(\sum_{j=1}^x z_j^{\lambda+})^{1/2}},
 \tag{15}$$

$$S_i^- = \frac{(\sum_{k=1}^x z_{ik}^{\lambda-})^{\frac{1}{2}} \cos \theta_{i-}}{(\sum_{j=1}^x z_j^{\lambda-})^{1/2}},
 \tag{16}$$

where

$$\cos \theta_{i+} = \frac{\sum_{j=1}^x z_{ij}^{\lambda+} z_j^{\lambda+}}{(\sum_{j=1}^x z_{ij}^{\lambda+} \sum_{j=1}^x z_j^{\lambda+})^{1/2}},
 \tag{17}$$

$$\cos \theta_{i-} = \frac{\sum_{j=1}^x z_{ij}^{\lambda-} z_j^{\lambda-}}{(\sum_{j=1}^x z_{ij}^{\lambda-} \sum_{j=1}^x z_j^{\lambda-})^{1/2}},
 \tag{18}$$

Step 9 Finally, the overall performance index for every alternative across all criteria may be calculated as

$$P_i = \frac{S_i^+}{S_i^+ + S_i^-}, \quad i= 1, 2, \dots, n.
 \tag{19}$$

Step 10 Using overall performance indexes, the ranking will be taken for all alternatives. Larger index value A_i is favored over the lesser index value alternative.

5. An empirical study

For software maintainability estimation three software products S1, S2, and S3 are taken. Software is graded according to factors of maintainability using similarity based approach. First, we will apply the method on sub goal analyzability, modifiability, stability, testability and understandability and then ranking of software for maintainability. Fuzzy AHP and TOPSIS are applies as combined approach. In Figure 3 there is a hierarchical structure for maintainability estimation. According to the maximum performance index it is concluded that S1 is much maintainable.

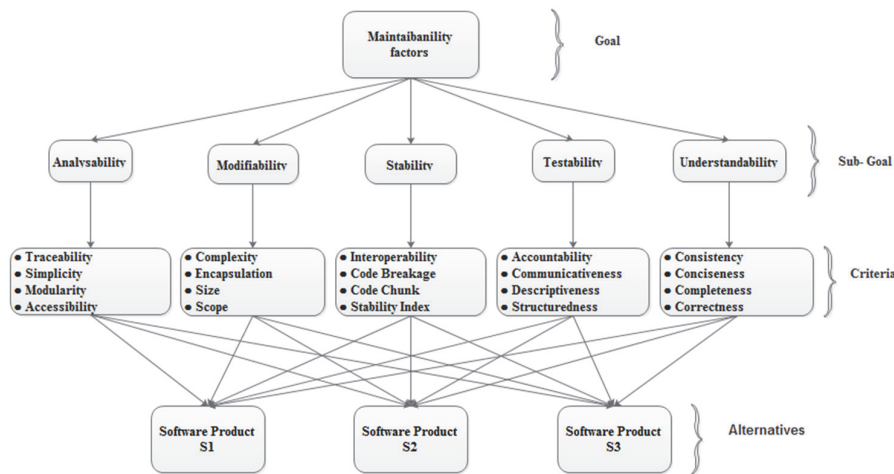


Fig. 3. Hierarchical Structure of maintainability estimation

5.1 Estimation of Analyzability

First, the method will be applied on analyzability and its factors. Here Analyzability is considered as the goal and traceability, simplicity, modularity and accessibility are taken as criteria, S1, S2 and S3 are the software product. In Fig. 4 a hierarchical construction of analyzability is presented. Here we will ranking the software based on analyzability by applying fuzzy multi-criteria analysis with similarity based approach.

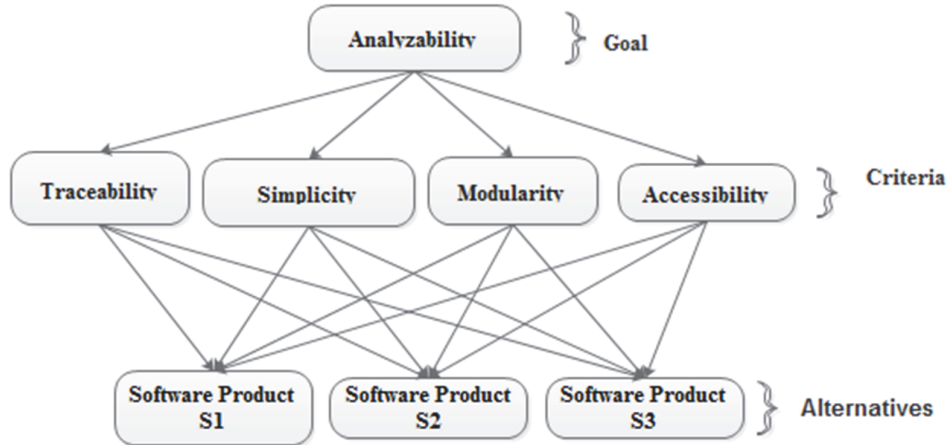


Fig. 4. Hierarchical Structure of analyzability estimation

Step 1 By using the expert advice for all criterions fuzzy judgment matrix generates. These values are provided by experts who are working in this field for previous five years. All these matrices are, given below.

$$C1 = \begin{bmatrix} 1 & 3 & 7 \\ 3^{-1} & 1 & 5 \\ 7^{-1} & 5^{-1} & 1 \end{bmatrix}, C2 = \begin{bmatrix} 1 & 3 & 5 \\ 3^{-1} & 1 & 7^{-1} \\ 5^{-1} & 7 & 1 \end{bmatrix}, C3 = \begin{bmatrix} 1 & 5 & 9^{-1} \\ 5^{-1} & 1 & 3 \\ 9 & 3^{-1} & 1 \end{bmatrix}, C4 = \begin{bmatrix} 1 & 7 & 5 \\ 7^{-1} & 1 & 3 \\ 5^{-1} & 3^{-1} & 1 \end{bmatrix}$$

By using the Table 1, each assessment of fuzzy judgment matrices are indicate in the figure of triangular fuzzy numbers (l, m, u) as follow.

$$C1 = \begin{bmatrix} (1,1,3) & (1,3,5) & (5,7,9) \\ (1/5,1/3,1) & (1,1,3) & (3,5,7) \\ (1/9,1/7,1/5) & (1/7,1/5,1/3) & (1,1,3) \end{bmatrix},$$

$$C2 = \begin{bmatrix} (1,1,3) & (1,3,5) & (5,7,9) \\ (1/5,1/3,1) & (1,1,3) & (1/9,1/7,1/5) \\ (1/7,1/5,1/3) & (5,7,9) & (1,1,3) \end{bmatrix},$$

$$C3 = \begin{bmatrix} (1,1,3) & (3,5,7) & (1/11,1/9,1/7) \\ (1/7,1/5,1/3) & (1,1,3) & (1,3,5) \\ (1/7,1/5,1/3) & (1/5,1/3,3) & (1,1,3) \end{bmatrix} \quad C4 = \begin{bmatrix} (1,1,3) & (5,7,9) & (3,5,7) \\ (1/9,1/7,1/5) & (1,1,3) & (1,3,5) \\ (1/7,1/5,1/3) & (1/5,1/3,1) & (1,1,3) \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 5 & 3 & 7 \\ 1/5 & 1 & 1/9 & 1/3 \\ 1/3 & 9 & 1 & 1/5 \\ 1/7 & 3 & 5 & 1 \end{bmatrix}$$

By applying Eq. (6) Decision matrix D will be calculated as

$$D1 = \begin{bmatrix} \frac{1+1+5}{1+1+5+1/5+1+3+1/9+1/7+1}, & \frac{1+3+7}{1+3+7+1/3+1+5+1/7+1/5+1}, & \frac{3+5+9}{3+5+9+1+3+7+1/5+1/3+3} \\ \frac{1+1+5+1/5+1+3+1/9+1/7+1}{1/5+1+3}, & \frac{1+3+7+1/3+1+5+1/7+1/5+1}{1/3+1+5}, & \frac{3+5+9+1+3+7+1/5+1/3+3}{1+3+7} \\ \frac{1+1+5+1/5+1+3+1/9+1/7+1}{1/9+1/7+1}, & \frac{1+3+7+1/3+1+5+1/7+1/5+1}{1/7+1/5+1}, & \frac{3+5+9+1+3+7+1/5+1/3+3}{1/5+1/3+3} \\ \frac{1+1+5+1/5+1+3+1/9+1/7+1}{1+1+5+1/5+1+3+1/9+1/7+1}, & \frac{1+3+7+1/3+1+5+1/7+1/5+1}{1+3+7+1/3+1+5+1/7+1/5+1}, & \frac{3+5+9+1+3+7+1/5+1/3+3}{3+5+9+1+3+7+1/5+1/3+3} \end{bmatrix}$$

Step 2 Similarly for C2, C3, C4 we write D2, D3, and D And finally we find the decision matrix (D) as

$$D = \begin{bmatrix} (0.562, 0.588, 0.539) & (0.401, 0.481, 0.475) & (0.539, 0.515, 0.444) & (0.722, 0.696, 0.602) \\ (0.337, 0.340, 0.348) & (0.105, 0.079, 0.133) & (0.282, 0.350, 0.365) & (0.169, 0.221, 0.261) \\ (0.100, 0.071, 0.112) & (0.493, 0.439, 0.391) & (0.177, 0.129, 0.189) & (0.107, 0.082, 0.137) \end{bmatrix}$$

Similarly weight (W) vector is written as

$$W = \begin{bmatrix} 0.401 & 0.428 & 0.437 \\ 0.057 & 0.044 & 0.081 \\ 0.335 & 0.282 & 0.279 \\ 0.205 & 0.244 & 0.364 \end{bmatrix},$$

Step 3 Now, by applying Eq. (7) the fuzzy performance matrix Z is calculating by using fuzzy arithmetic.

$$Z = \begin{bmatrix} (0.225, 0.251, 0.235) & (0.022, 0.021, 0.038) & (0.180, 0.145, 0.123) & (0.148, 0.169, 0.219) \\ (0.135, 0.145, 0.152) & (0.005, 0.003, 0.010) & (0.094, 0.098, 0.101) & (0.034, 0.053, 0.094) \\ (0.040, 0.030, 0.048) & (0.028, 0.019, 0.031) & (0.059, 0.036, 0.052) & (0.021, 0.020, 0.049) \end{bmatrix}$$

Step 4 By applying Eq. (9), for this we using a = 0.5, then the interval performance is attained as shown below.

$$Z_a = \begin{bmatrix} (0.238, 0.227) & (0.021, 0.029) & (0.0162, 0.134) & (0.158, 0.194) \\ (0.140, 0.149) & (0.004, 0.006) & (0.096, 0.099) & (0.043, 0.073) \\ (0.035, 0.039) & (0.006, 0.025) & (0.047, 0.044) & (0.020, 0.034) \end{bmatrix}$$

Step 5 Final crisp performance matrix is considered by using optimism index $\lambda = 0.5$ as practical view

$$Z_a^{\lambda} = \begin{bmatrix} 0.232 & 0.024 & 0.0148 & 0.176 \\ 0.144 & 0.005 & 0.097 & 0.094 \\ 0.194 & 0.015 & 0.095 & 0.197 \end{bmatrix},$$

Step 6 For obtaining normalized performance matrix, we need to normalize the crisp performance matrix by using Eq. (12) as follow.

$$Z_a^{\lambda} = \begin{bmatrix} 0.692 & 0.828 & 0.736 & 0.628 \\ 0.429 & 0.173 & 0.482 & 0.335 \\ 0.579 & 0.517 & 0.472 & 0.704 \end{bmatrix},$$

Step 7 Finding the positive and negative ideal solutions, by applying Eq. (14) on normalized performance matrix.

$$A_a^{\lambda^+} = (0.692 \quad 0.828 \quad 0.736 \quad 0.704),$$

$$A_a^{\lambda^-} = (0.429 \quad 0.173 \quad 0.472 \quad 0.335),$$

Step 8 We find the degree of similarity between each alternative for each positive and negative ideal solution by using Eq. (15)–(16) as shown below.

$$S_1^+ = 0.975, S_2^+ = 0.466, S_3^+ = 0.757, \\ S_1^- = 1.093, S_2^- = 0.605, S_3^- = 0.794,$$

Step 9 Finally, the ranking of all alternative may be decided by using performance index as follow. P1=0.470, P2=0.299, P3=0.488

Step 10 Table 3 shows the outcome of all alternatives position by using performance index. According to the result, grading the first choice order of these software is S3, S1 and S2. Therefore, the best analyzability estimation is S3.

Table 3
Ranking of software for analyzability using performance index

Soft-ware	Performance index	Ranking
S1	0.470	2
S2	0.299	3
S3	0.488	1

5.2 Estimation of modifiability

Similarly, performance index for modifiability is calculated according the alternatives and criteria as shown in Fig. 5.

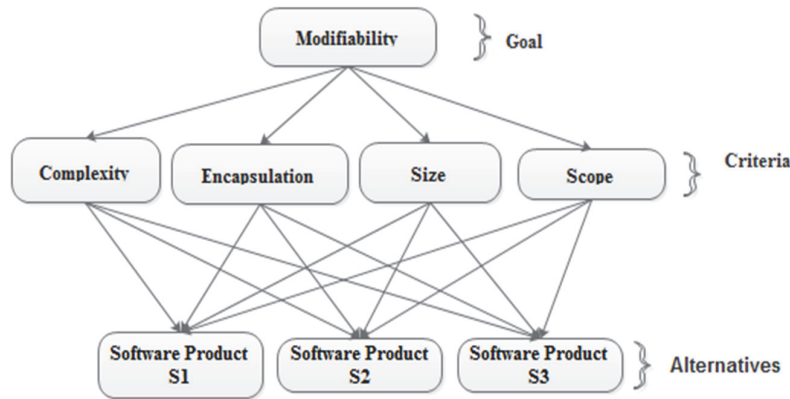


Fig. 5. Hierarchical Structure of modifiability

Step 1 Fuzzy judgment matrix generates are given below.

$$C1 = \begin{bmatrix} 1 & 3 & 5 \\ 3^{-1} & 1 & 7 \\ 5^{-1} & 7^{-1} & 1 \end{bmatrix}, C2 = \begin{bmatrix} 1 & 3 & 7 \\ 3^{-1} & 1 & 5 \\ 7^{-1} & 5^{-1} & 1 \end{bmatrix}, C3 = \begin{bmatrix} 1 & 9 & 5^{-1} \\ 9^{-1} & 1 & 3 \\ 5 & 3^{-1} & 1 \end{bmatrix}, C4 = \begin{bmatrix} 1 & 5 & 7 \\ 5^{-1} & 1 & 3^{-1} \\ 7^{-1} & 3 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 3 & 7 & 1/3 \\ 1/3 & 1 & 5 & 1/9 \\ 1/7 & 1/5 & 1 & 1/5 \\ 3 & 9 & 5 & 1 \end{bmatrix}$$

Step 2 Similarly for C2, C3, C4 we write D2, D3, and D And finally we find the decision matrix (D) as

$$D = \begin{bmatrix} (0.401,0.482,0.475) & (0.562,0.589,0.539) & (0.333,0.494,0.428) & (0.722,0.697,0.539) \\ (0.497,0.446,0.412) & (0.338,0.339,0.348) & (0.085,0.199,0.243) & (0.107,0.082,0.137) \\ (0.100,0.071,0.112) & (0.101,0.071,0.112) & (0.172,0.306,0.328) & (0.169,0.221,0.260) \end{bmatrix}$$

Similarly weight (W) vector is written as

$$W = \begin{bmatrix} 0.289 & 0.303 & 0.295 \\ 0.172 & 0.172 & 0.182 \\ 0.056 & 0.041 & 0.063 \\ 0.482 & 0.482 & 0.458 \end{bmatrix}$$

Step 3 By applying Eq. (7) the fuzzy performance matrix Z is finding using fuzzy arithmetic.

$$Z = \begin{bmatrix} (0.116,0.146,0.140) & (0.096,0.101,0.099) & (0.019,0.020,0.026) & (0.348,0.335,0.246) \\ (0.143,0.135,0.122) & (0.059,0.059,0.063) & (0.004,0.008,0.015) & (0.051,0.039,0.062) \\ (0.029,0.022,0.033) & (0.017,0.122,0.021) & (0.009,0.012,0.020) & (0.081,0.106,0.119) \end{bmatrix}$$

Step 4 By considering Eq. (9), for this we taking $\alpha = 0.5$, then the interval performance is attained as given below.

$$Z_{\alpha} = \begin{bmatrix} (0.131,0.143) & (0.962,0.100) & (0.019,0.023) & (0.341,0.290) \\ (0.139,0.128) & (0.059,0.034) & (0.006,0.011) & (0.045,0.050) \\ (0.025,0.027) & (0.069,0.121) & (0.010,0.016) & (0.093,0.112) \end{bmatrix}$$

Step 5 Final crisp performance matrix is considered by using optimism index $\lambda = 0.5$ as reasonable view.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.065 & 0.531 & 0.021 & 0.315 \\ 0.133 & 0.046 & 0.008 & 0.047 \\ 0.026 & 0.004 & 0.013 & 0.102 \end{bmatrix}$$

Step 6 For obtaining normalized performance matrix, normalizing the crisp performance matrix by using Eq. (12) as follow.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.432 & 0.996 & 0.807 & 0.943 \\ 0.885 & 0.086 & 0.307 & 0.141 \\ 0.173 & 0.007 & 0.500 & 0.305 \end{bmatrix}$$

Step 7 Finding the positive and negative ideal solutions, by applying Eq. (14) on normalized performance matrix.

$$A_{\alpha}^{\lambda+} = (0.885 \quad 0.996 \quad 0.807 \quad 0.943),$$

$$A_{\alpha}^{\lambda-} = (0.173 \quad 0.007 \quad 0.307 \quad 0.141),$$

Step 8 We find the degree of similarity between each software for each positive and negative ideal solution by using Eq. (15)–(16) as shown below.

$$S_1^+ = 0.878, S_2^+ = 0.217, S_3^+ = 0.256, \\ S_1^- = 3.197, S_2^- = 1.893, S_3^- = 1.567,$$

Step 9 Finally, the ranking of all alternative may be decided as follow (by using performance index).
 $P1=0.215, P2=0.102, P3=0.140$

Step 10 Table 4 shows the results of all software and grading by considering performance index. According to the result, ranking the first selection order of software is S1, S3 and S2. Therefore, the best analyzability estimation is S2.

Table 4
 Ranking of software for modifiability using performance index

Software	Performance index	Ranking
S1	0.215	1
S2	0.102	3
S3	0.140	2

5.3 Estimation of Stability

Similarly, performance index for stability is calculated according the alternatives and criteria as shown in Fig. 6.

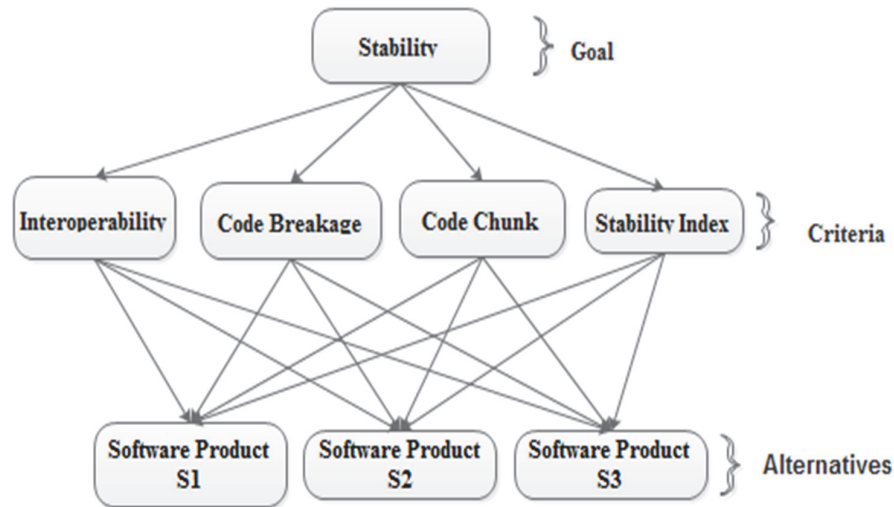


Fig. 6. Hierarchical Structure of Stability

Step 1 The fuzzy judgment matrix generates are given as below.

$$C1 = \begin{bmatrix} 1 & 5 & 9 \\ 5^{-1} & 1 & 7 \\ 9^{-1} & 7^{-1} & 1 \end{bmatrix}, C2 = \begin{bmatrix} 1 & 5 & 7 \\ 5^{-1} & 1 & 3 \\ 7^{-1} & 3^{-1} & 1 \end{bmatrix}, C3 = \begin{bmatrix} 1 & 3 & 9 \\ 3^{-1} & 1 & 7^{-1} \\ 9^{-1} & 7 & 1 \end{bmatrix}, C4 = \begin{bmatrix} 1 & 3 & 7 \\ 3^{-1} & 1 & 5^{-1} \\ 7^{-1} & 5 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 5 & 7 & 1/3 \\ 1/5 & 1 & 3 & 9 \\ 1/7 & 1/3 & 1 & 1/5 \\ 3 & 1/9 & 5 & 1 \end{bmatrix}$$

Step 2 The decision matrix (D). The fuzzy judgment matrix (W) as

$$D = \begin{bmatrix} (0.599,0.613,0.572) & (0.790,0.696,0.602) & (0.548,0.575,0.537) & (0.562,0.588,0.539) \\ (0.334,0.335,0.336) & (0.188,0.224,0.264) & (0.079,0.065,0.118) & (0.107,0.082,0.137) \\ (0.065,0.051,0.091) & (0.115,0.079,0.133) & (0.371,0.359,0.343) & (0.330,0.328,0.323) \end{bmatrix}$$

Similarly weight (W) vector is written as

$$W = \begin{bmatrix} 0.369 & 0.357 & 0.338 \\ 0.367 & 0.353 & 0.327 \\ 0.058 & 0.044 & 0.076 \\ 0.204 & 0.244 & 0.375 \end{bmatrix}$$

Step 3 Now, by applying Eq. (7) the fuzzy performance matrix Z is finding using fuzzy arithmetic.

$$Z = \begin{bmatrix} (0.221,0.218,0.193) & (0.289,0.245,0.196) & (0.031,0.025,0.040) & (0.114,0.143,0.202) \\ (0.123,0.119,0.113) & (0.068,0.079,0.086) & (0.004,0.002,0.008) & (0.021,0.020,0.051) \\ (0.023,0.018,0.030) & (0.042,0.027,0.043) & (0.021,0.015,0.026) & (0.067,0.080,0.121) \end{bmatrix}$$

Step 4 By taking Eq. (9), for this we using a = 0.5, then the interval performance is attained as given below.

$$Z_a = \begin{bmatrix} (0.219,0.205) & (0.267,0.220) & (0.028,0.032) & (0.128,0.172) \\ (0.121,0.116) & (0.073,0.075) & (0.003,0.005) & (0.020,0.035) \\ (0.020,0.024) & (0.034,0.035) & (0.018,0.020) & (0.073,0.101) \end{bmatrix}$$

Step 5 Final crisp performance matrix is given as below.

$$Z_a^{\lambda'} = \begin{bmatrix} 0.212 & 0.243 & 0.030 & 0.150 \\ 0.118 & 0.111 & 0.003 & 0.027 \\ 0.022 & 0.034 & 0.019 & 0.087 \end{bmatrix}$$

Step 6 After normalizing the crisp performance matrix we obtain normalized performance matrix as follow.

$$Z_a^{\lambda} = \begin{bmatrix} 0.870 & 0.902 & 0.857 & 0.857 \\ 0.484 & 0.413 & 0.085 & 0.154 \\ 0.090 & 0.126 & 0.542 & 0.497 \end{bmatrix}$$

Step 7 Finding the positive and negative ideal solutions, by applying Eq. (14) on normalized performance matrix.

$$A_a^{\lambda^+} = (0.870 \quad 0.902 \quad 0.867 \quad 0.857)$$

$$A_a^{\lambda^-} = (0.090 \quad 0.126 \quad 0.085 \quad 0.154)$$

Step 8 The positive and negative ideal solution is calculated as below.

$$S_1^+ = 0.999, S_2^+ = 0.327, S_3^+ = 0.355$$

$$S_1^- = 7.194, S_2^- = 2.281, S_3^- = 2.651$$

Step 9 Finally performance index is calculated as below.

P1=0.455, P2=0.125, P3=0.118

Step 10 Table 5 shows the result of all software grading by using performance index. According to the result, ranking the first selection order of these alternatives is S2, S1 and S3. Therefore, the best analyzability estimation is S2.

Table 5
Ranking of software for stability using performance index

Software	Performance index	Ranking
S1	0.302	2
S2	0.405	1
S3	0.235	3

5.4 Estimation of Testability

Similarly, performance index for testability is calculated according the alternatives and criteria as shown in Fig. 7.

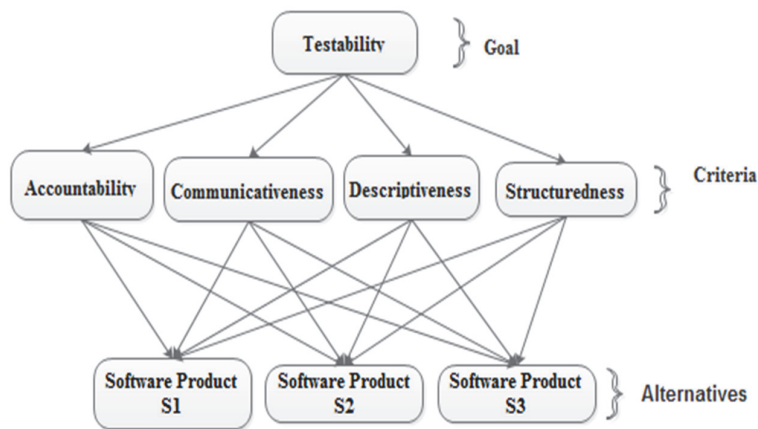


Fig. 7. Hierarchical Structure of Testability

Step 1 The fuzzy judgment matrix given below.

$$C1 = \begin{bmatrix} 1 & 3 & 9 \\ 3^{-1} & 1 & 5 \\ 9^{-1} & 5^{-1} & 1 \end{bmatrix}, C2 = \begin{bmatrix} 1 & 5 & 9 \\ 5^{-1} & 1 & 7^{-1} \\ 9^{-1} & 7 & 1 \end{bmatrix}, C3 = \begin{bmatrix} 1 & 3^{-1} & 7 \\ 3 & 1 & 5 \\ 7^{-1} & 5^{-1} & 1 \end{bmatrix}, C4 = \begin{bmatrix} 1 & 5 & 3^{-1} \\ 5^{-1} & 1 & 9 \\ 3 & 9^{-1} & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 3 & 7 & 5 \\ 1/3 & 1 & 1/9 & 1/3 \\ 1/7 & 9 & 1 & 7 \\ 1/5 & 3 & 1/7 & 1 \end{bmatrix}$$

Step 2 The decision matrix (D). The fuzzy judgment matrix (W) as

$$D = \begin{bmatrix} (0.623,0.590,0.540) & (0.599,0.613,0.572) & (0.498,0.447,0.412) & (0.291,0.307,0.328) \\ (0.291,0.339,0.349) & (0.069,0.054,0.097) & (0.402,0.482,0.475) & (0.564,0.494,0.428) \\ (0.085,0.071,0.187) & (0.249,0.332,0.331) & (0.101,0.072,0.112) & (0.145,0.199,0.243) \end{bmatrix}$$

Similarly weight (W) vector is written as

$$W = \begin{bmatrix} 0.372 & 0.407 & 0.394 \\ 0.055 & 0.045 & 0.084 \\ 0.488 & 0.638 & 0.381 \\ 0.083 & 0.111 & 0.140 \end{bmatrix}$$

Step 3 Now, the fuzzy performance matrix Z is finding as below.

$$Z = \begin{bmatrix} (0.231,0.240,0.212) & (0.032,0.027,0.048) & (0.243,0.285,0.156) & (0.024,0.034,0.045) \\ (0.108,0.138,0.138) & (0.003,0.003,0.008) & (0.196,0.307,0.180) & (0.046,0.055,0.059) \\ (0.031,0.028,0.073) & (0.014,0.015,0.027) & (0.008,0.045,0.043) & (0.012,0.022,0.034) \end{bmatrix}$$

Step 4 The interval performance is attained using $\alpha = 0.5$ as given below.

$$Z_{\alpha} = \begin{bmatrix} (0.235,0.226) & (0.029,0.037) & (0.264,0.220) & (0.029,0.039) \\ (0.125,0.138) & (0.003,0.005) & (0.251,0.243) & (0.050,0.057) \\ (0.029,0.050) & (0.014,0.021) & (0.026,0.044) & (0.017,0.028) \end{bmatrix},$$

Step 5 Final crisp performance matrix is calculated by using optimism index $\lambda = 0.5$ as reasonable view.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.230 & 0.033 & 0.242 & 0.034 \\ 0.131 & 0.003 & 0.247 & 0.053 \\ 0.039 & 0.017 & 0.035 & 0.022 \end{bmatrix}$$

Step 6 By using normalizing the crisp performance matrix we obtain normalized performance matrix, by using Eq. (12) as follow.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.861 & 0.887 & 0.697 & 0.507 \\ 0.490 & 0.081 & 0.712 & 0.798 \\ 0.146 & 0.457 & 0.101 & 0.335 \end{bmatrix}$$

Step 7 Finding the positive and negative ideal solutions, by applying Eq. (14) on normalized performance matrix.

$$A_{\alpha}^{\lambda+} = (0.861 \quad 0.887 \quad 0.712 \quad 0.798)$$

$$A_{\alpha}^{\lambda-} = (0.146 \quad 0.081 \quad 0.101 \quad 0.335)$$

Step 8 Find the degree of similarity between each alternative for each positive and negative ideal solution by using Eq. (15)–(16) as shown below.

$$S_1^+ = 0.907, \quad S_2^+ = 0.612, \quad S_3^+ = 0.324, \\ S_1^- = 2.895, \quad S_2^- = 2.763, \quad S_3^- = 1.198,$$

Step 9 The performance index for the ranking of all software is calculated as follow.

P1=0.238, P2=0.181, P3=0.212

Step 10 Table 6 shows the result of all software ranking by using performance index. According to the result, ranking the first selection order of these alternatives is S1, S3 and S2. Therefore, the best Testability estimation is S1.

Table 6

Ranking of software for testability using performance index

Software	Performance index	Ranking
S1	0.238	1
S2	0.181	3
S3	0.212	2

5.5 Estimation of Understandability

Similarly, performance index for understandability is calculated according the alternatives and criteria as shown in Fig. 8.

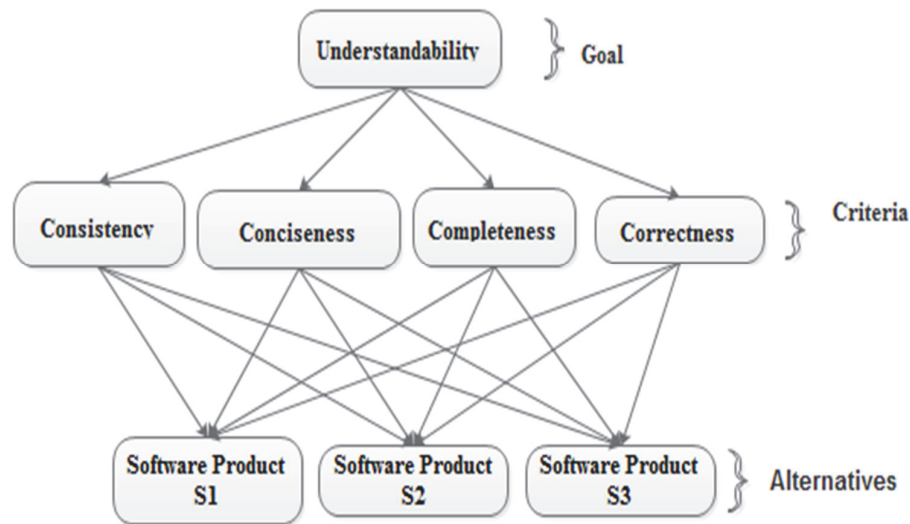


Fig. 8. Hierarchical Structure of understandability

Step 1 The fuzzy judgment matrix are given below.

$$C1 = \begin{bmatrix} 1 & 5 & 9^{-1} \\ 5^{-1} & 1 & 3^{-1} \\ 9 & 3 & 1 \end{bmatrix}, C2 = \begin{bmatrix} 1 & 3 & 9 \\ 3^{-1} & 1 & 5^{-1} \\ 9^{-1} & 5 & 1 \end{bmatrix}, C3 = \begin{bmatrix} 1 & 7^{-1} & 3^{-1} \\ 7 & 1 & 9 \\ 3 & 9^{-1} & 1 \end{bmatrix}, C4 = \begin{bmatrix} 1 & 5^{-1} & 3 \\ 5 & 1 & 7 \\ 3^{-1} & 7^{-1} & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 3 & 7 & 5 \\ 1/3 & 1 & 1/9 & 1/3 \\ 1/7 & 9 & 1 & 7 \\ 1/5 & 3 & 1/7 & 1 \end{bmatrix}$$

Step 2 The decision matrix (D). The fuzzy judgment matrix (W) as

$$D = \begin{bmatrix} (0.283,0.296,0.303) & (0.623,0.629,0.567) & (0.064,0.065,0.134) & (0.172,0.224,0.264) \\ (0.093,0.074,0.129) & (0.093,0.074,0.129) & (0.833,0.752,0.606) & (0.723,0.696,0.602) \\ (0.623,0.629,0.567) & (0.283,0.296,0.302) & (0.102,0.182,0.259) & (0.105,0.079,0.133) \end{bmatrix}$$

Similarly weight (W) vector is written as

$$W = \begin{bmatrix} 0.372 & 0.407 & 0.394 \\ 0.055 & 0.045 & 0.084 \\ 0.488 & 0.638 & 0.381 \\ 0.083 & 0.111 & 0.140 \end{bmatrix}$$

Step 3 The fuzzy performance matrix Z is finding using fuzzy arithmetic as below.

$$Z = \begin{bmatrix} (0.105,0.120,0.119) & (0.034,0.028,0.047) & (0.031,0.041,0.015) & (0.014,0.024,0.036) \\ (0.034,0.030,0.051) & (0.005,0.003,0.011) & (0.406,0.479,0.230) & (0.060,0.077,0.084) \\ (0.231,0.256,0.223) & (0.015,0.013,0.025) & (0.049,0.116,0.098) & (0.008,0.088,0.018) \end{bmatrix}$$

Step 4 The interval performance is attained by using $\alpha = 0.5$, then as given below.

$$Z_{\alpha} = \begin{bmatrix} (0.112,0.119) & (0.031,0.037) & (0.036,0.028) & (0.019,0.030) \\ (0.032,0.040) & (0.004,0.007) & (0.442,0.354) & (0.068,0.080) \\ (0.243,0.239) & (0.014,0.019) & (0.082,0.107) & (0.048,0.053) \end{bmatrix}$$

Step 5 By using optimism index $\lambda = 0.5$ the crisp performance matrix is calculated as below.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.116 & 0.034 & 0.032 & 0.024 \\ 0.036 & 0.005 & 0.398 & 0.074 \\ 0.241 & 0.016 & 0.094 & 0.050 \end{bmatrix}$$

Step 6 For obtaining normalized performance matrix, normalizing the crisp performance matrix by using Eq. (12) as follow.

$$Z_{\alpha}^{\lambda} = \begin{bmatrix} 0.431 & 0.918 & 0.078 & 0.260 \\ 0.132 & 0.135 & 0.970 & 0.804 \\ 0.895 & 0.432 & 0.229 & 0.543 \end{bmatrix}$$

Step 7 Finding the positive and negative ideal solutions, by applying Eq. (14) on normalized performance matrix.

$$A_{\alpha}^{\lambda+} = (0.895 \quad 0.918 \quad 0.970 \quad 0.804),$$

$$A_{\alpha}^{\lambda-} = (0.132 \quad 0.135 \quad 0.078 \quad 0.260),$$

Step 8 Find the scale of similarity between each alternative for each positive and negative ideal solution by using Eq. (15)–(16) as shown below.

$$S_1^+ = 0.467, S_2^+ = 0.565, S_3^+ = 0.276,$$

$$S_1^- = 2.310, S_2^- = 6.989, S_3^- = 3.070,$$

Step 9 Finally, the performance index of all software will decide as follow.

$$P1=0.433, P2=0.143, P3=0.326$$

Step 10 Table 7 shows the result of all alternatives ranking by using performance index. According to the result, grading the first choice order of these alternatives is S1, S3 and S2. Therefore, the best understandability estimation is S1.

Table 7

Ranking of software for understandability using performance index

Soft-ware	Performance index	Ranking
S1	0.101	1
S2	0.092	3
S3	0.049	2

5.6 Estimation of Maintainability

In this section, maintainability performance index has been estimated by normalizing the performance indexes of analyzability, modifiability, stability, testability and understandability. Normalization of performance indexes is carried out by taking the average of all these five factors. Table 8 presents the overall ranking of software for maintainability by using performance index of all factors. According to the result, ranking the first choice order of these alternatives is S1, S3 and S2. Therefore, the best maintainability estimation is S1.

Table 8

Ranking of software for maintainability using performance index

Soft-ware	Performance index	Ranking
S1	0.362	1
S2	0.170	3
S3	0.256	2

5.7 Discussions

A visualization of comparative ranking within factors (analyzability, modifiability, stability, testability and understandability) from data of Table 3 to Table 8, a comparative graph is generated as shown in Figure 9. This graph infer that software S3 is best for analyzability, S1 is best for modifiability, S2 is best for stability, S1 is best for testability, S1 is best for understandability. Therefore, the best maintainability estimation is S1. A novel approach to rank the software for maintainability estimation is presented in this thesis. Maintainability estimation of software by ranking them based on their performance index resulted from operating a soft computing technique has emerged as a new concept in this area. This approach is based on factors assessments that affect the software maintainability. As we know there are number of factors that affect the software maintainability. In this research work we have identified and classified critical factors that affect the software maintainability along with their relationship to the software maintainability. Accordingly we present a hierarchy of factors to determine the best alternative for maintainability prediction by using multiple criteria analysis approach. We proposed the approach by which we have the best decision that promises greater maintainability of software. This approach provides a new direction for software maintainability based on hierarchy of factors assessment is present in literature, so comparison is not given. Importantly, this research work offered

a significant benefit of the implementation process that are more readily identified, analyzed and a satisfactory study undertaken.

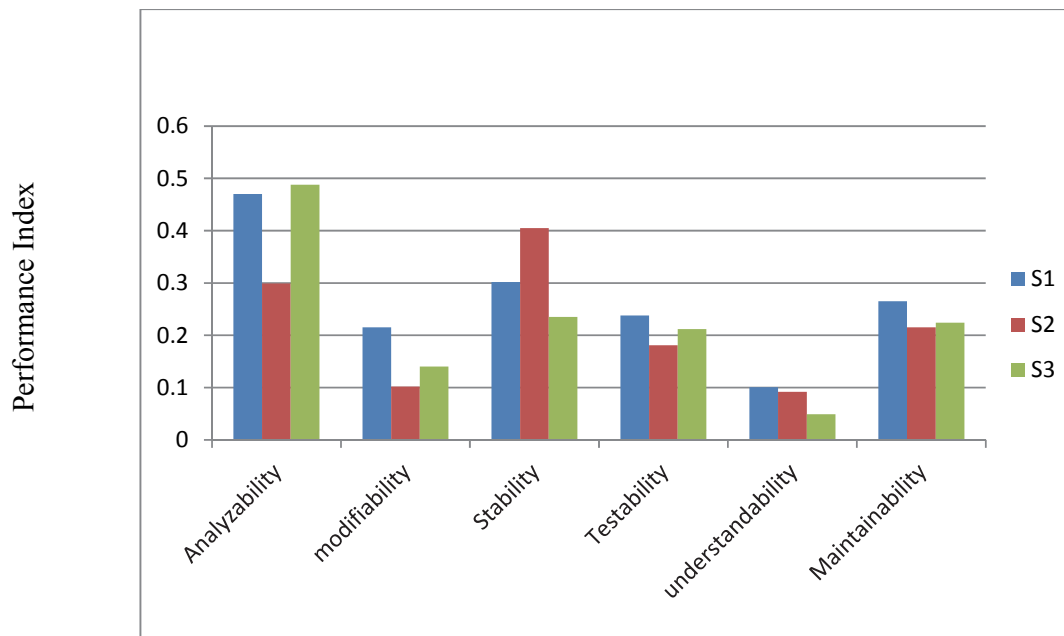


Fig. 9. Visualization of software ranking

6. Conclusion

In this paper, the MA analysis has been used for similarity based approach and proposed a new method for maintainability estimation. A similarity-based approach using fuzzy AHP and TOPSIS has been used at 2-level hierarchy to evaluate the identified significant factors that affect maintainability. Weight determination is a challenging task in decision making problems, in this paper, the pair-wise comparison matrix has been employed with fuzzy elements for this purpose. On the basis of generated performance index by TOPSIS, a ranking of candidate software is established for maintainability estimation. Maximum performance index is calculated for each sub goal and then for major goal. The approach is illustrated by using the example.

Acknowledgement

The first author would like to thank Dr. Kirti Seth, Associate Professor, CSE Department, A.K.G.E.C., Ghaziabad, whose timing supervision and guidance has provided the moral boost in doing this work, answering a number of technical queries despite her busy schedule.

References

- Cai, L., Liu, Z., Zhang, J., Tong, W., & Yang, G. (2010, August). Evaluating Software Maintainability Using Fuzzy Entropy Theory. In *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on* (pp. 737-742). IEEE.
- Chen, J., & Liu, X. (2009, December). Software maintainability metrics based on the index system and fuzzy method. In *Information Science and Engineering (ICISE), 2009 1st International Conference on* (pp. 5117-5120). IEEE.
- Chidamber, S. R., & Kemerer, C. F. (1991). Towards a metrics suite for object oriented design. *ACM*, 26(11), 197-211.
- Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.

- Deng, H. (1999, August). Multicriteria analysis with fuzzy pairwise comparison. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International* (Vol. 2, pp. 726-731). IEEE.
- Deng, H. (2007, August). A similarity-based approach to ranking multicriteria alternatives. In *International Conference on Intelligent Computing* (pp. 253-262). Springer Berlin Heidelberg.
- Dubey, S. K., & Rana, A. (2012). A Fuzzy Approach for Evaluation of Maintainability of Object Oriented Software System. *International Journal of Computer Applications*, 49(21).
- Fulei, W. (2010). The research on college teacher performance evaluation based on fuzzy-AHP method. *In: Proceedings of the 2nd international workshop on education technology and computer science, IEEE computer society*.
- Gani, A. N., & Assarudeen, S. M. (2012). A new operation on triangular fuzzy number for solving fuzzy linear programming problem. *Applied Mathematical Sciences*, 6(11), 525-532.
- Ghosh, S., & Dubey, S. K. (2012). Fuzzy Maintainability Model for Object Oriented Software System.
- Goel, N., Dubey, S. K., & Rana, A. (2012). Fuzzy layered approach for maintainability evaluation of object oriented software system. *International Journal of Scientific & Engineering Research*, 3(6), 1.
- Gyimothy, T., Ferenc, R., & Siket, I. (2005). Empirical validation of object-oriented metrics on open source software for fault prediction. *IEEE Transactions on Software engineering*, 31(10), 897-910.
- Hwang, C.L., & Yoon, K.P. (1981). *Multiple attributes decision making methods and applications*. Springer, Berlin, pp 12-21
- Jeet, K., & Dhir, R. (2012). Bayesian and fuzzy approach to assess and predict the maintainability of software: A comparative study. *ISRN Software Engineering, 2012*.
- Kumar, V., Sharma, A., Kumar, R., & Grover, P. S. (2012). Quality aspects for component-based systems: A metrics based approach. *Software: Practice and Experience*, 42(12), 1531-1548.
- Li, W. (1998). Another metric suite for object-oriented programming. *Journal of Systems and Software*, 44(2), 155-162.
- Mahmood, Z. (1982). *Multiple criteria decision making*. McGraw-Hill, New York
- Momeni, H., & Zahedian, S. (2014). Aspect-oriented software maintainability assessment using Adaptive Neuro Fuzzy Inference System (ANFIS).
- Moradi, M., & Ebrahimi, E. (2014). Applying Fuzzy AHP and Similarity-Based Approach for Economic Evaluating Companies Based on Corporate Governance Measures. *Global Journal of Management Studies and Researches*, 1(1), 10-20.
- Seetha lakshmi V (2013) Decision making in academic institutions: A fuzzy ahp approach. *International Journal of Engineering Science and Technology*, 1(5), 403-409.
- Saaty, T.L. (1980). *The Analytic Hierarchy Process*. McGraw-Hill, New York.
- Zhang, L. (2010). Comparison of classical analytic hierarchy process (AHP) approach and fuzzy AHP approach in multiple-criteria decision making for commercial vehicle information systems and networks (CVISN) project.

