# Preferential Selection of Software Quality Models Based on a Multi-Criteria Decision-Making Approach

Ankita Verma, Jaypee Institute of Information Technology, Noida, India
Anushka Agarwal, Jaypee Institute of Information Technology, Noida, India
Manisha Rathore, Jaypee Institute of Information Technology, Noida, India
Sneha Bisht, Jaypee Institute of Information Technology, Noida, India
Deepti Singh, Jaypee Institute of Information Technology, Noida, India*

## ABSTRACT

Software engineering mainly aims to produce software of good quality that is delivered on time and on budget. Software quality becomes an important concern for quantifying the performance of software attributes. The seminal objective of the work is to choose the appropriate software quality model according to the client's needs where the client can give more importance to specific criteria compared to others as per his/her application's requirements. The proposed approach will help to decide the best alternative suitable for the application. The work is based on selecting the most suitable software quality model taking all the parameters into consideration while making the decision using multi-criteria decision-making techniques.

## KEYWORDS

Fuzzy Logic, Multi-Criteria Decision Making, Software Quality Model

## 1. INTRODUCTION

Software engineering is the application of a systematic, disciplined, and quantifiable approach to the development, operation and maintenance of software (Bourque & Dupuis 2004) Software development and evolution is characterized by multiple objectives and constraints (Ruhe, 2002) Thus, in order to develop software that caters to the needs of the client, it becomes inevitable to assess its quality. The quality of a software can be assessed based on various parameters by using standard software quality models (Al-Badareen et.al, 2011). However, difficulty arises when the client wishes to consider multiple aspects at once and not just one criterion for deciding the most apt model for their application. It is difficult to decide the most suitable software quality model based on all of the client's requirements. The work presented here is focused on choosing the best software quality

DOI: 10.4018/IJSI.315739 *Corresponding Author

model according to the client's needs where the client can give more importance to specific criteria compared to others as per his/her application requirements. Then it will help them decide the best alternative suitable for their application. Since, this task is complex and depended upon multiple factors, so we applied multi-criteria decision making (MCDM) approaches.

Multi-Criteria Decision Making (MCDM) techniques provide solutions to the problems involving conflicting and multiple objectives (Velasquez & Hester, 2013). It has all the characteristics of a useful decision support tool. MCDM problems are comprised of five components: Goal, Decision maker or group of decision makers with opinions (preferences), Decision alternatives, Evaluation criteria (interests), Outcomes or consequences associated with alternative/interest combination. For our proposed work, the goal is to find the most suitable software quality model. This decision can be made by the software developers on the basis of the preferences provided by the clients. Various software attributes such as reliability, correctness, reusability etc. will act as the evaluation criteria. The set of alternatives that will be evaluated in our work are: McCall (McCall et.al, 1977), ISO-9126 (Behkamal et.al, 2009), FURPS (Al-Qutaish, 2010), Boehm (Boehm et.al, 1978) and Dromey (Dromey, 1995), The work proposed in this article is based on the application of three Multi Criteria Decision making algorithms namely Analytic Hierarchy Process, TOPSIS and PROMETHEE II along with the application of Fuzzy Logic [10,11].

The rest of the paper is organized as follows: the next section highlights few related works in the literature. Section 3 explains our proposed work. In Section 4, the experimental analysis is included. Section 5 concludes the article with future directions.

## 2. RELATED WORK

Multi-Criteria Decision Making has been extensively employed in various fields of study as they are well suited to handle the inherent complexity of real-life decision making. With the emergence of novel technologies and improvements in the existing ones, MCDM has expanded its scope of applicability (Velasquez & Hester, 2013). MCDM approaches have found their application for various tasks in software engineering as well. Measuring software quality is one of most important aspect of a software development process. To this end, many models have been proposed in the literature (McCall et.al, 1977, Behkamal et.al, 2009, Al-Qutaish, 2010, Bohem, 1978, Dromey, 1995). Al Badreen (2011) has proposed a formal framework conduct an in-depth evaluation of competing software quality models. They have attempted to quantify the performance of the various software quality models such as McCall (McCall et.al, 1977), ISO-9126 (Behkamal et.al, 2009), FURPS (Al-Qutaish, 2010), Boehm (Boehm et.al, 1978), and Dromey (Dromey, 1995) based on several factors such as reliability, efficiency, integrity etc. In (Sehra, 2016), the authors have presented various fields of application of MCDM algorithms in the domain of software engineering. Apart from this, they have also compared the performance of AHP and fuzzy AHP to choose the software quality model based on only three criteria reliability, efficiency and maintainability. In their study, the software quality models that were considered are McCall (McCall et.al, 1977), ISO-9126 (Behkamal et.al, 2009), and Boehm (Boehm et.al, 1978) In (Jusoh et.al, 2014), the authors have applied AHP algorithm for selecting the appropriate Open-Source Software for the development of the project. They have started with the goal of selecting new database for the product development. The three alternatives: mySQL, PostGreSQL, FireBird are evaluated on the basis of various parameters such as system reliability, system usability, system functionality, system performance, information security etc., to name few. In similar line of work, MCDM approaches Fuzzy AHP and Topsis are successfully employed for selecting appropriate Software Development Life Cycle model (Khan et.al, 2014). They have taken into consideration three SDLC models: Traditional, Agile, and Hybrid approaches. The criteria over which these models were evaluated are based on requirements clarity, development time and cost, requirements change, system complexity, stakeholder communication and development team size. In (Challa et.al, 2011), the authors have attempted the different perspectives of users, managers and developers for assessing quality of the software using MCDM approaches.

## 3. PROPOSED APPROACH

In our work, we proposed for applying various multi-criteria decision making approaches for the task of finding the most appropriate software quality model based on various criteria. In this section, we will start by discussing the employed MCDM approaches. Then, the chosen set of alternatives, the criteria and the application steps will be presented.

### 3.1 Methodologies Used

#### *3.1.1 Fuzzy Analytic Hierarchy Process (Fuzzy AHP)*

In our work, fuzzy AHP (Jing et.al, 2018) is applied on the dataset to find the relationship between different criteria and to assign a certain significant value to each and every criterion which further aided in other algorithms used such as Fuzzy TOPSIS and PROMETHEE.

The basic steps of fuzzy AHP algorithm are mentioned below:

**Step 1:** A $n \times n$ comparison matrix is created with the linguistic terms and then it is converted to the corresponding fuzzy numbers on Saaty's 1–9 scale as shown in Table 1. The element $\tilde{a}_{ij}$ can be interpreted as the degree of preference of the $i^{th}$ criterion over the $j^{th}$ criterion:

$$A = \begin{bmatrix} 1 & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & 1 \end{bmatrix} \quad (1)$$

where:

$$\tilde{a}_{ij} = \begin{cases} \left(\frac{1}{p_i}, \frac{1}{m_i}, \frac{1}{n_i}\right) & \forall i < j \\ (1,1,1) & i = j \\ (p_i, m_i, n_i) & \forall i > j \end{cases}$$

**Table 1. Preference Scale between Two Components in AHP**

| Scales | Degree of Preferences | Fuzzy Triangular Number |
|---|---|---|
| 1 | Equally | (1,1,1) |
| 3 | Moderately | (1,3,5) |
| 5 | Strongly | (3,5,7) |
| 7 | Very Strongly | (5,7,9) |
| 9 | Extremely | (7,9,9) |

Note: Values show relative importance of one component as compared to other component. Reciprocals are used for inverse comparison.

**Step 2:** Geometric mean is then obtained for every row of $A$ using the geometric mean technique as given below multiplied by the fuzzy relative weight matrix is deduced by equation 3:

$$\tilde{r}_i = \left(\tilde{a}_{i1} \otimes \ldots \otimes \tilde{a}_{ij} \otimes \ldots \otimes \tilde{a}_{in}\right)^{1/n} \forall i = 1,2,3,\ldots n \tag{2}$$

$$\tilde{w}_i = \tilde{r}_i \oplus \left(\tilde{r}_1 \otimes \ldots \otimes \tilde{r}_i \oplus \ldots \otimes \tilde{r}_n\right)^{-1} \forall i = 1,2,3,\ldots n \tag{3}$$

where $\tilde{w}_i$ gives the final weight value of $i^{th}$ criterion, $\otimes$ is the fuzzy numbers product and $\oplus$ is the fuzzy numbers sum.

### *3.1.2 Fuzzy Technique for Order of Preference by Similarity Ideal Solution (F-TOPSIS)*

FTOPSIS one of the Multi Criteria Decision Making (MCDM) analysis method. It provides a solution such that it is closest to the best ideal solution and farthest from the worst ideal solution (Hwang & Yoon, 1992).

The basic steps of fuzzy Topsis can be explained below:

**Step 1:** The decision makers who would give their linguistic (fuzzy) judgement (rankings) to various available alternatives (solutions) for each criterion. Convert the responses obtained in the decision matrices to a fuzzy scale using the conversion in Table 2.

**Step 2:** Assuming there are $K$ decision makers, so we will have $K$ such decision matrices. The next step is to obtain a Combined Decision Matrix using equation 4. The fuzzy rating of $i^{th}$ alternative for $j^{th}$ criterion is given by ($a_{ij}$, $b_{ij}$, $c_{ij}$) such that:

$$a_{ij} = \min_k \left\{a_{ij}^k\right\}, b_{ij} = \frac{1}{K}\sum_k b_{ij}^k, c_{ij} = \max_k \left\{c_{ij}^k\right\} \tag{4}$$

**Step 3:** Next step is to normalize the decision matrix which comprises of the benefit criteria values and cost criteria values. The benefit criteria are those criteria are those which maximum value is desired. Non benefit or cost criteria are those criteria which minimum value is desired.

For benefit criterion:

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*}\right), c_j^* = \max_i \left\{c_{ij}\right\} \tag{5}$$

**Table 2. Linguisting responses with corresponding fuzzy numbers**

| Responses | Fuzzy Triangular Number |
|---|---|
| Very low | (1,1,1) |
| Low | (1,3,5) |
| Moderate | (3,5,7) |
| High | (5,7,9) |
| Very High | (7,9,9) |

For cost criterion:

$$\tilde{r}_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}}\right), a_j^- = \min_i \left\{a_{ij}\right\} \tag{6}$$

**Step 4:** After that, we form a weighted normalized decision matrix V by multiplying the normalized decision matrix $R$ obtained in the previous step with the weights assigned to each criterion using Fuzzy AHP (obtained in previous section):

$$\tilde{v}_{ij} = \tilde{r}_{ij} \times \tilde{w}_j \tag{7}$$

**Step 5:** Compute the Fuzzy Positive Ideal Solution (FPIS) and Fuzzy Negative Ideal Solution (FNIS) represented by A* and $A^-$ respectively:

$$A^* = \left(\tilde{v}_1^*, \tilde{v}_2^*, \tilde{v}_3^*, \ldots, \tilde{v}_n^*\right), \tilde{v}_j^* = \max_i \left\{v_{ij3}\right\} \tag{8}$$

$$A^- = \left(\tilde{v}_1^-, \tilde{v}_2^-, \tilde{v}_3^-, \ldots, \tilde{v}_n^-\right), \tilde{v}_j^- = \min_i \left\{v_{ij1}\right\} \tag{9}$$

**Step 6:** Compute the distances $\left(d_i^*\right.$ and $\left.d_i^-\right)$ for each of the $m$ alternatives to the FPIS and FNIS as:

$$d_i^* = \sum\nolimits_{j=1}^{n} d\left(\widetilde{v_{ij}}, \tilde{v}_j^*\right) \tag{10}$$

$$d_i^- = \sum\nolimits_{j=1}^{n} d\left(\widetilde{v_{ij}}, \tilde{v}_j^-\right) \tag{11}$$

where:

$$d\left(\tilde{x}, \tilde{y}\right) = \sqrt{\frac{1}{3}\left[\left(a_1 - a_2\right)^2 + \left(b_1 - b_2\right)^2 + \left(c_1 - c_2\right)^2\right]}$$

**Step 7:** Compute the Closeness Coefficient ($CC_i$) of each alternative to rank the alternatives in decreasing order of $CC_i$ value which is computed as:

$$CC_i = \frac{d_i^-}{\left(d_i^- + d_i^*\right)} \tag{12}$$

### *3.1.3 PROMETHEE II*

This algorithm will recommend the best alternative according to the preferences of criteria and the corresponding weights of these criteria. This algorithm will help people to take the correct decisions in terms of their preferred criteria (Murat et.al, 2015). The foundation of this approach is a comparison pair for each pair of potential alternatives along each criterion. Different criteria are used to analyse potential alternatives and must either be maximised or minimised. Each criterion must have a weight and a preference function. For each of the alternative, a preference index is computed based on the net outrank flow which signifies the dominance of the concerned alternative over the other alternatives.

The basic steps of PROMETHE-II algorithm are given as below:

**Step1:** Decision matrix is normalized as per the beneficial and non-beneficial attributes using the given formula.

Beneficial criterion:

$$R_{ij} = \left[X_{ij} - \min_j(X_{ij})\right] \Big/ \left[\max_j X_{ij} - \min_j X_{ij}\right] \tag{13}$$

Non-beneficial criterion:

$$R_{ij} = \left[\max_j(X_{ij}) - X_{ij}\right] \Big/ \left[\max_j X_{ij} - \min_j X_{ij}\right] \tag{14}$$

where $X_{ij}$ is the metric representing performance of $i^{th}$ alternative in $j^{th}$ criterion.

**Step 2:** Calculate the preference function $P_j\left(i,i'\right)$ for each of the $j^{th}$ criterion and each pairwise alternatives $i \ and \ i'$ given as:

$$P_j\left(i,i'\right) = \begin{cases} 0 \ if \ R_{ij} \le R_{i'j} \\ \left(R_{ij} - R_{i'j}\right) if \ R_{ij} > R_{i'j} \end{cases} \tag{15}$$

**Step 3:** For each of the $j^{th}$ criterion, compute the aggregated preference function $\pi\left(i,i'\right)$ by using the given formula:

$$\pi\left(i,i'\right) = \frac{\sum_{j=1}^{m} w_j P_j\left(i,i'\right)}{\sum_{j=1}^{m} w_j} \tag{16}$$

**Step 4:** Leaving and entering outranking flows are computed using equations 16 and 17 respectively. For $i^{th}$ alternative, the leaving flow represents domination of this attribute the other n-1 alternatives, while the entering flow expresses the degree by which this attribute is dominated by others:

$$\varphi^{+}\left(i\right)=\frac{1}{n-1}\sum_{i'=1}^{n}\pi\left(i,i'\right)i\neq i' \tag{17}$$

$$\varphi^{-}\left(i\right)=\frac{1}{n-1}\sum_{i'=1}^{n}\pi\left(i',i\right)i\neq i \tag{18}$$

Net outranking flow $\varphi\left(i\right)$ for $i^{th}$ alternative:

$$\varphi\left(i\right)=\varphi^{+}\left(i\right)-\varphi^{-}\left(i\right) \tag{19}$$

Final ranking of alternatives is computed based of ϕ(i). The higher value of ϕ(i), the better is the alternative.

### 3.2 Criteria

Criteria for quality model selection are based on the research conducted till now. Various latest research papers were referred for the selection of criteria. The criteria chosen are:

- **Reliability:** It is the quality of the system to operate without failure in the given environment for a given period (R. Mall., 2004).
- **Maintainability:** Once the software is delivered to the client, faults might occur in its components during operation. The ability of the system of identify and mitigate faults is referred to as maintainability (R. Mall., 2004).
- **Efficiency:** It is the ability of the software to optimally use the system resources such as computational time, disk space, memory, network etc., for its function (R. Mall., 2004).
- **Portability:** It characterizes the capability of the software to well adapt to the changes in the operational environment or modifications in the requirements (R. Mall., 2004).
- **Integrity:** It is concerned about the software's capability of protection against illegal access and attacks (R. Mall., 2004).
- **Testability:** It is the degree measures the ease of fault detection during testing of the software (R. Mall., 2004).

### 3.3 Alternatives

List of chosen Software Quality Model as alternatives:

- **McCall:** McCall's Quality Model (McCall, 1977) is one of the earlier proposed models which attempts to consider the perspectives of users' as well as developers' in accessing software quality. All the attributes are also arranged in the following broad categories: Product Operation (Correctness, Reliability, Efficiency, Integrity, Usability) Product Transition (Portability, Reusability, Interoperability) and Product Revision (Maintainability, Flexibility, Testability).
- **ISO 9126:** ISO/IEC 9126 standard (Behkamal, 2009) lays primary focus on the following six characteristics Functionality, Reliability, Usability, Efficiency, Maintainability, Portability in the process of software quality assessment.
- **FURPS: (**FURPS, 2010) stands for following five characteristics: Functionality, Usability, Reliability, Performance, Supportability.
- **Boehm: (**Boehm et.al, 1978) quality model aims to automatically quantify the quality of defined set of attributes and metrics. Briefly, the major software attributes focused are: Reliability, Portability, Human Engineering, Testability, Efficiency, Understandability, Modifiability.

- **Dromey: (**Dromey, 1995) proposed a product based quality model that advocates the use of dynamic quality estimation model depending upon the characteristics of the products. Overall, the focus is on the mentioned these primary software product characteristics: Functionality, Reliability, Maintainability, Efficiency, Reliability, Maintainability, Reusability, Portability, Reliability, Maintainability, Efficiency, Reliability, Usability.

### 3.4 Implementation

The basic steps followed in our study are as follows:

- The software quality models that will be compared are selected as alternatives.
- The software quality attributes based on which the alternatives will be assessed are decided as criteria based on literature survey.
- The weights of the criteria are computed by applying FuzzyAHP based on the relative preferences given by the decision makers. Here, the client as well as the developer can participate in decision-making process.
- The decision matrix, which specifies the rating of each alternative for all the criteria is constructed, based on the input from the decision makers (developers/tester). For our empirical study, we have taken the rating from the literature (Al-Badareen, 2011) as shown in Table 3.
- Fuzzy TOPSIS and PROMETHE-II are applied to obtain the rankings of the alternatives based on user's preferences.

Figure 1 shows the overall approach diagrammatically.

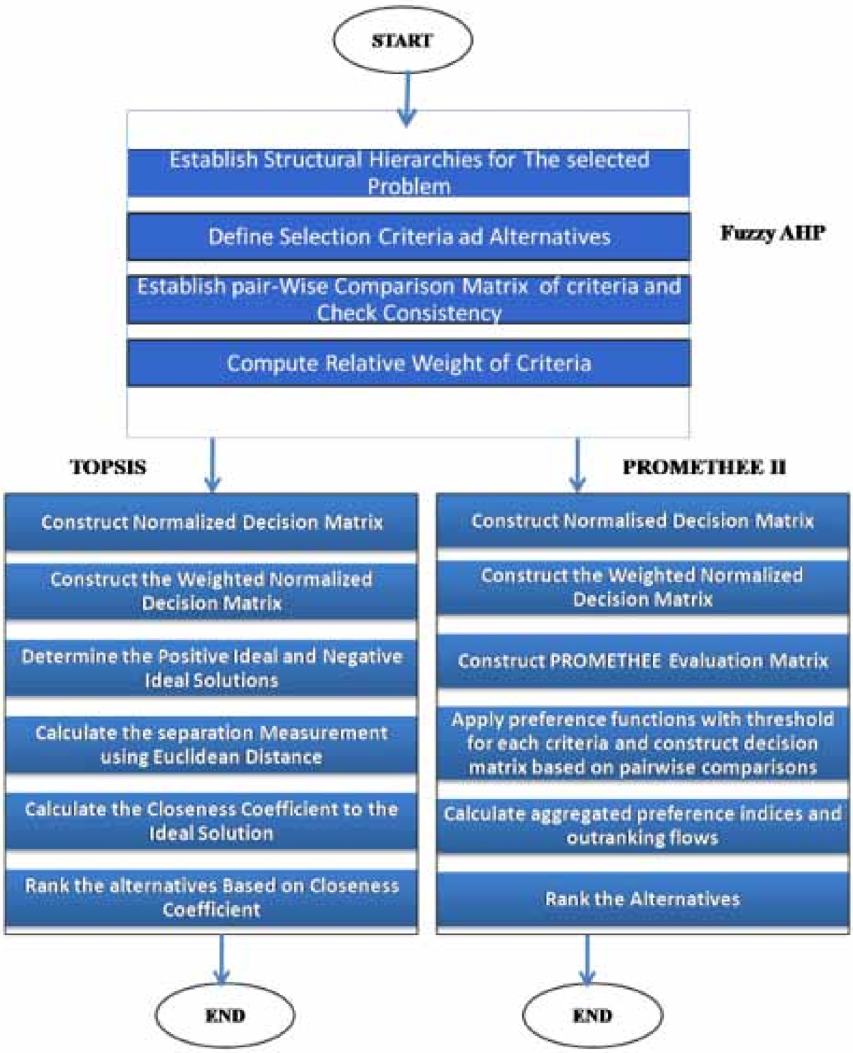
## 4. RESULTS AND DISCUSSION

Results are analysed on the basis of the ranks obtained from PROMETHEE-II and Fuzzy TOPSIS for varying inputs so as to compare the results and sensitivity of the result to the input data. For each of the given input, we have plotted the weight-age of criteria, the net ranking obtained by the alternatives from PROMETHE-II and the closeness coefficient obtained by Fuzzy TOPSIS for the alternatives. The results of two such inputs are shown in Figure 2 and 3. We have obtained results after giving different input values and obtained different ranks, as shown in the above table 4. This shows that our output varies according to input. Also, we have obtained the results from two algorithms, which provide us similar results thus giving more belief into its accuracy. It can be seen from the table that when equal weightage is given to all the criteria, the ranks obtained by both the MCDM approaches matches with the ranks obtained in (Al-Badareen et.al, 2011). However, it does not take into consideration, the relative importance of various criteria as per the choice of the selector, therefore

**Table 3. Ratings of the selected attributes (Al-Badareen et.al, 2011)**

| | McCall | ISO 9126 | Boehm | FURPS | Dromey |
|---|---|---|---|---|---|
| **Reliability** | 6.5 | 7 | 5 | 6.5 | 5 |
| **Maintainability** | 6.8 | 7.3 | 6.4 | 0.5 | 5 |
| **Efficiency** | 7 | 8 | 5.5 | 2.5 | 5 |
| **Testability** | 7.8 | 2.5 | 5.3 | 0.5 | 0.5 |
| **Portability** | 6.7 | 7.8 | 6.1 | 1 | 5.5 |
| **Integrity** | 9.5 | 2.5 | 2.5 | 2.5 | 1.5 |

**Figure 1. Overall Approach**




for second input the ranks obtained by both the approaches changed. This highlights the fact that the developer can choose different software quality model by assigning different weight-age to different factors as per his requirement.

## 5. CONCLUSION

In this article, MCDM based approach is proposed to select appropriate software quality model based on client's preferences. Taking a close look at the problem discussed, we can say that it is expandable to other domains as well. However, there are certain limitations in the proposed method such as to establish the validity of the approach is difficult as the values entered by the client may change after a period of time. In future, there is a possibility of achieving results that are even more accurate by taking more criteria and models into consideration.

**Figure 2. Input1 a) Weights of criteria b) Net outrank of PROMETHEE -II c) Closeness Coefficient of Fuzzy TOPSIS**

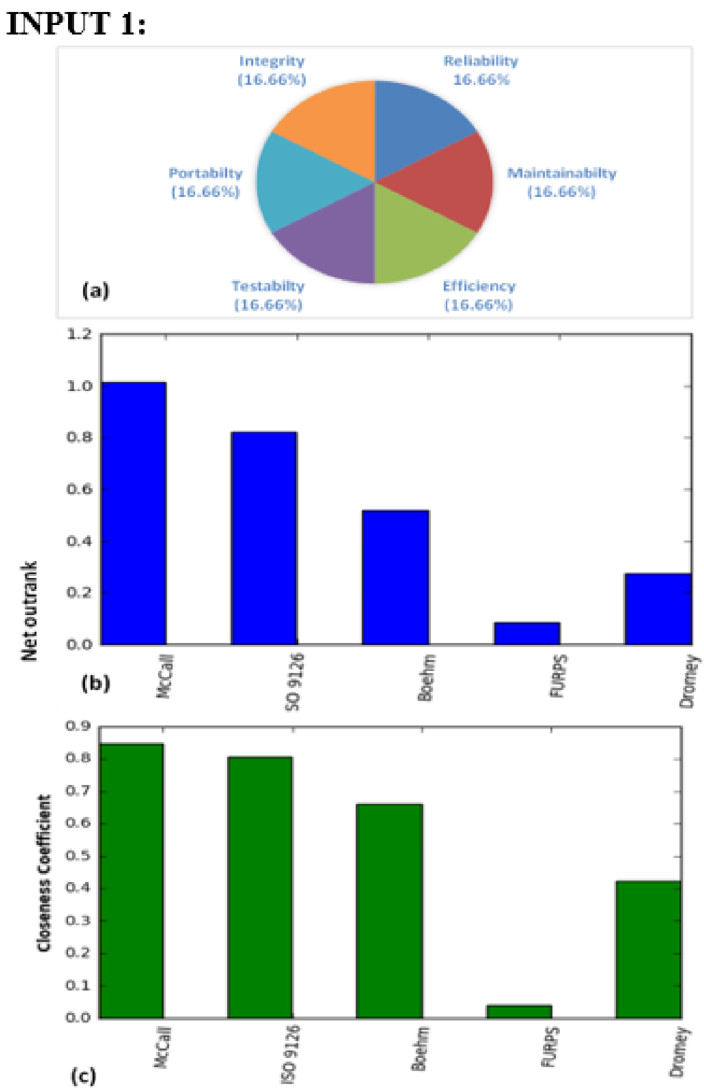


Figure 3. Input1 a) Weights of criteria b) Net outrank of PROMETHEE -II c) Closeness Coefficient of Fuzzy TOPSIS

**INPUT 2:**

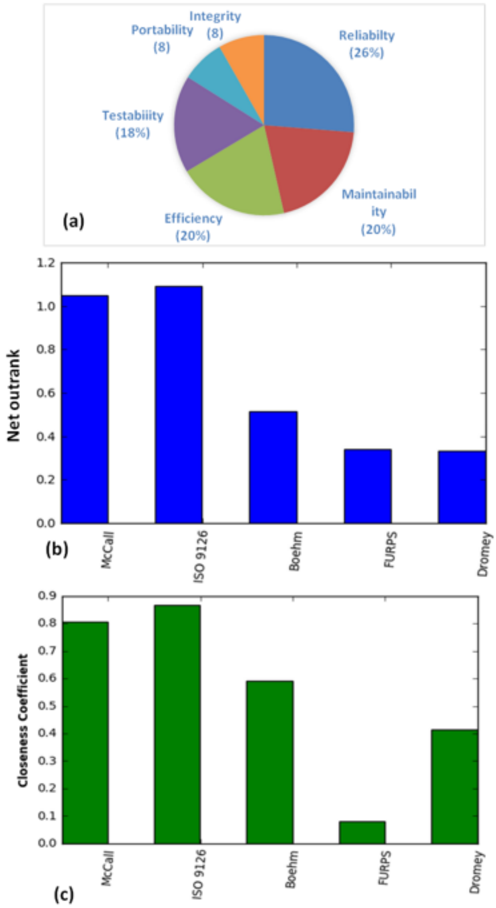



Table 4. Ranks obtained by different methods

| Models | INPUT 1 | | INPUT 2 | | Rank(Al-Badareen et.al, 2011) |
|---|---|---|---|---|---|
| | PROMETHEE-II | FTOPSIS | PROMETHEE-II | FTOPSIS | |
| **McCall** | 1 | 1 | 2 | 2 | 1 |
| **ISO 9126** | 2 | 2 | 1 | 1 | 2 |
| **Boehm** | 3 | 3 | 3 | 3 | 3 |
| **FURPS** | 5 | 5 | 4 | 5 | 5 |
| **Dromey** | 4 | 4 | 5 | 4 | 4 |

## REFERENCES


Al-Badareen, A. B., Selamat, M. H. A., Jabar, M., Din, J., & Turaev, S. (2011). Software quality models: A comparative study. In *International Conference on Software Engineering and Computer Systems* (pp. 46-55). Springer. doi:10.1007/978-3-642-22170-5_4

Al-Qutaish, R. E. (2010). Quality models in software engineering literature: An analytical and comparative study. *The Journal of American Science*, *6*(3), 166–175.

Behkamal, B., Kahani, M., & Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and Software Technology*, *51*(3), 599–609. doi:10.1016/j.infsof.2008.08.001

Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., & MacLeod, G. (1978). *Merritt: Characteristics of Software Quality*. Academic Press.

Bourque, P., & Dupuis, R. (2004). Guide to the software engineering body of knowledge 2004 version. *Guide to the Software Engineering Body of Knowledge, 2004. SWEBOK*, 12.

Challa, J. S., Paul, A., Dada, Y., Nerella, V., Srivastava, P. R., & Singh, A. P. (2011). Integrated software quality evaluation: A fuzzy multi-criteria approach. *Journal of Information Processing Systems*, *7*(3), 473–518. doi:10.3745/JIPS.2011.7.3.473

Chen, S. J., & Hwang, C. L. (1992). Fuzzy multiple attribute decision making methods. In *Fuzzy multiple attribute decision making* (pp. 289–486). Springer. doi:10.1007/978-3-642-46768-4_5

Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on Software Engineering*, *21*(2), 146–162. doi:10.1109/32.345830

Jing, M., Jie, Y., Shou-yi, L., & Lu, W. (2018). Application of fuzzy analytic hierarchy process in the risk assessment of dangerous small-sized reservoirs. *International Journal of Machine Learning and Cybernetics*, *9*(1), 113–123. doi:10.1007/s13042-015-0363-4

Jusoh, Y. Y., Chamili, K., Pa, N. C., & Yahaya, J. H. (2014). Open source software selection using an analytical hierarchy process (AHP). *American Journal of Software Engineering and Applications*.

Kahraman, C. (Ed.). (2008). *Fuzzy multi-criteria decision making: theory and applications with recent developments* (Vol. 16). Springer Science & Business Media. doi:10.1007/978-0-387-76813-7

Khan, M. A., Parveen, A., & Sadiq, M. (2014, February). A method for the selection of software development life cycle models using analytic hierarchy process. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)* (pp. 534-540). IEEE. doi:10.1109/ICICICT.2014.6781338

Mall, R. (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd.

McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in software quality. Volume I. Concepts and definitions of software quality*. General Electric.

Murat, S., Kazan, H., & Coskun, S. S. (2015). An application for measuring performance quality of schools by using the PROMETHEE multi-criteria decision making method. *Procedia: Social and Behavioral Sciences*, *195*, 729–738. doi:10.1016/j.sbspro.2015.06.344

Ruhe, G. (2002). Software engineering decision support–a new paradigm for learning software organizations. In *International Workshop on Learning Software Organizations* (pp. 104-113). Springer.

Saaty, T. L., & Vargas, L. G. (2001). How to make a decision. In *Models, methods, concepts & applications of the analytic hierarchy process* (pp. 1–25). Springer. doi:10.1007/978-1-4615-1665-1_1

Sehra, S. K., Brar, Y. S., & Kaur, N. (2016). Applications of multi-criteria decision making in software engineering. *International Journal of Advanced Computer Science and Applications*, *7*(7), 472–477.

Velasquez, M., & Hester, P. T. (2013). An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, *10*(2), 56–66.

*Ankita Verma is currently working as an Assistant Professor in Jaypee Institute of Information Technology, Noida. She has received her Ph.D. and M.Tech degrees from Jawaharlal Nehru University, New Delhi in the year 2018 and 2014 respectively. She has published in reputed journal and conferences. Her research interests include Delay Tolerant Network, Social Network Analysis, Computational Intelligence and Nature Inspired Computing.*