World Applied Sciences Journal 32 (4): 718-730, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.32.04.1102

Testability Assessment of Aspect Oriented Software Using Multicriteria Decision Making Approaches

[1]*Pradeep Kumar Singh*, [2]*Om Prakash Sangwan*,
[1]*Amrendra Pratap and* [1]*Amar Pal Singh*

[1]Amity University Uttar Pradesh, Noida, India
[2]Gautam Buddha University, Greater Noida, India

Abstract: Testability is an essential criterion for software quality and it is always difficult to measure. Multicriteria Decision Making (MCDM) approaches are very effectively and widely used in predicting the quality of the software. MCDM techniques are very helpful in ambiguous range of decisions. It has the ability to grip the uncertainty of pairwise comparisons. In this paper, multicriteria decision making approach has been used for qualitative assessment of aspect oriented software. MCDM approaches applied in this paper for testability predictions are Analytical Hierarchy Process (AHP), Fuzzy AHP (FAHP) and Preference Ranking Organization Method of Enrichment Evaluations (PROMETHEE-2). These approaches used for decision making through uncertainty of expert decisions. A pairwise comparison has been made based on expert judgements to a certain extent than exact numerical values. Four aspect oriented programs have been compared based on five quality attributes of software testability. Initially, the validations of software testability factors have been made through AHP. Ranking of programs are made through AHP, fuzzy AHP and PROMETHEE-2 approaches. Results show that the applied approaches are efficient and proved its suitability for the prediction of the testability of aspect oriented programs.

Key words: Software Testability · Aspect Oriented Programs (AOP) · Aspect Oriented Software Development (AOSD) · Multi Criteria Decision Making (MCDM) · Controllability · Observability, Built-in-test Capability · Understandability and Complexity.

INTRODUCTION

Testability is one of the qualitative factors of software engineering and ISO has defined software testability as functionality. It defines functionality as "the collection of characteristics of software that bear on the effort required to authenticate the software produced" [1].

IEEE defines it as "an activity in which a component or a system is evaluated for some specific conditions, the results are examined and evaluation is based on some aspect of the component or the system" [2].

It is also well known reality that more than 50% of the total cost in the development of software is related to the software testing activities [3]. Hence, in software development life cycle, testing is the most expensive phase in terms of efforts needed, money as well as time. So, it is very important to reduce the efforts and time required for testing the software's.

The testability of software components can be determined by factors such as-

Controllability: The degree to which it is possible to control all values of its individual output domain.

Observability: The degree to which it is possible to observe accurate output for a specified input.

Built-in-Test Capability: It has the ability to test the software itself. It reduces the complexity as well as decreases the cost of software. It can improve controllability and observability.

Understandability: The degree to which the component under test is documented or self-explaining.

Corresponding Aurthor: Pradeep Kumar Singh, Amity University Uttar Pradesh, Noida, India.

Complexity: It is the quantitative measurement of the complexity of the program. Low complexity of any software system is an indication of high quality.

The Multicriterion Decision Making (MCDM) methods are widely used for the evaluation of complex real-world problems. It has the ability to judge different alternatives on various criteria for selection of the best suitable alternatives. Experts decisions are evaluated through pairwise comparisons in AHP, however due to uncertainty they cannot give crisp values. Thus, AHP process joined with fuzzy set theory for better prediction [4]. Stewart [5], analyzed the MCDM process and needfulness in future as (a) the practical justification and analysis of the different available approaches (b) the expansion of MCDM for a set of decision making conditions and (c) dealing with uncertainty.

There are some mathematical models which are used to simplify MCDM method are analytical hierarchy process (AHP), Fuzzy-AHP and preference ranking organization method of enrichment evaluations (PROMETHEE-2). These methods are possible to implement in solving logistic as well as technical systems. More research work is still need to shrink the space between practical and theory in MCDM approach for the possibility to solve realistic problems. Duckstein *et al.* [6], prepared the procedural steps for the multicriteria approach to choose best suitable solution from available choices. It can be expressed as:

- Identifying the criteria for the main problem.
- Collection of proper data.
- Organization of realistic/proficient alternatives.
- Preparing the payoff matrix.
- Choosing the proper way to solve the problem.
- Integration of decision maker's preference arrangement.
- Selecting the best suitable alternatives for the analysis.

We have taken four alternatives from AspectJ programs i.e. Observer (P_1), Bean (P_2), Marketing (P_3) and Progress Monitor (P_4) evaluated on the basis of five criteria which are Controllability (F_1), Observability (F_2), Built in test capability (F_3), Understandability (F_4) and Complexity (F_5) to access the software testability, whose relationship is shown in Figure 1. These five factors for testability have been already analyzed and their relationship has been reported in our previous work [9].

The detail of Aspect Oriented Programs is shown in Table 1, as shown below:

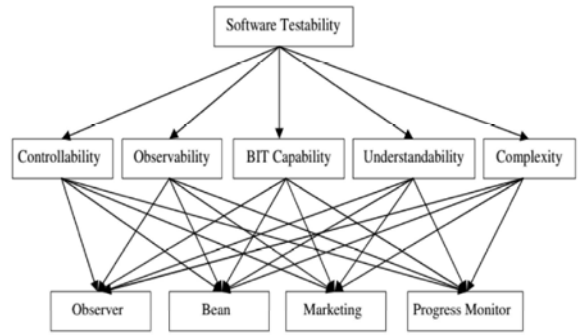


Fig. 1: Decision Making Levels

Table 1: AspectJ Programs Details

	TLOC	TNOC	TNOM	TNOP	TCC	CHE
Observer (P_1)	72	6	13	1	13	8915
Bean (P_2)	78	2	11	1	13	15147
Marketing (P_3)	145	5	22	1	27	18684
Progress Monitor (P_4)	227	6	29	2	32	26467

Where, TLOC, TNOC, TNOM, TNOP, TCC and CHE are defined as total lines of code, total no. of classes, total no. of methods, total number of packages, total cyclomatic complexity and cumulative Halstead effort respectively. We have already evaluated these four programs in term of reusability using MCDM approaches in our previous work [25]. However, this paper emphasize on testability assessment using the MCDM approached for AO programs.

This paper is divided in five sections. In first section, introduction to software testability and multicriteria approaches are discussed. In second part, the related work based on testability of software's and multicriteria approach used in the prediction of quality of software is considered. In next section, testability model and applied approaches are described. In fourth section, analysis of AspectJ programs using the MCDM approaches is carried out. Finally, conclusion in context of considered aspect oriented programs and their testability is drawn along with future scope.

Related Works: Till now, very few studies which are related to multicriteria decision making approach has been published in context to software engineering problems. Some of the important theories determine that the decision is defined as choice of the most suitable alternative with respect to predefined criteria [7]. In [8], Saaty proposed AHP as one of the most practical method based on MCDM. Laarhoven *et al.* [4] combined classical AHP with fuzzy set theory. Stewart [5] analyzed the MCDM process and their needfulness in future. Tomić *et al.* [10]

discussed PROMETHEE method with multicriteria decision making for four alternatives for choosing best suitable ones. In [11], Behzadian *et al.* introduces a complete literature study on methodologies and applications based on PROMETHEE approach.

In addition to MCDM, there are some theories in which testability is taken as an important factor for maintainability.

In [12], Challa *et al.* proposed a quality model in which maintainability is one of the factor of software quality model and sub-characteristics of it as analyzability, changeability, testability, stability, maintainability compliance and track-ability. In [13], Singh and Sangwan proposed a model based on maintainability for aspect oriented software. In [14], Kumar describes maintainability as a factor in which sub-characteristics are analyzability, changeability, testability, stability and modularity. In [15], Santos *et al.* proposed maintainability for aspect oriented software implementation model in which analyzability, changeability, testability and stability are taken as a sub-characteristics of maintainability. In [16], Rønningen *et al.* describes six quality characteristics of the ISO 9126 in which maintainability is one of the factor and further divided in sub-characteristics as analyzability, changeability, testability, stability and adaptability. In [17] Sharma *et al.* gives a maintainability model and a comparative analysis of sub-attributes which are testability, understandability, modifiability, stability, analyzability, changeability, conciseness, self descriptiveness, modularity, compliance and simplicity. In [18], Ghosh *et al.* gives a comparative study of maintainability factors in which testability, modifiability, stability, analyzability, changeability, modularity, adaptability and many more sub-characteristics of maintainability are taken.

Based on review work mentioned above, it can be concluded that testability is a sub-attribute of maintainability and important quality characteristics for software. To evaluate the testability of software, MCDM approaches may also be applied for qualitative and quantitative assessment. Based on MCDM, best alternatives can be chosen, while selecting software among multiple available choices in context to testability. So, we have considered four aspects programs to choose their testability based on MCDM approaches for analysis.

MATERIALS AND METHDS

As we have identified five factors to assess testability for aspect oriented programs. A relationship between identified factors and software testability is shown in Figure 2. Testability is considered as the main sub-characteristic of maintainability in several studies. However, in order to evaluate testability as a main and an important sub-characteristic, we have applied a multicriteria decision making approach on the proposed model as show below:

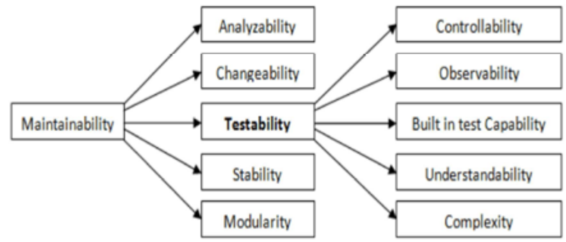


Fig. 2: Software Testability Model for AO Software

Table 2: The Saaty Rating Scale [8]

Intensityof importance	Definition	Explanation
1	The same importance	Objective is donated by two factors equally.
3	Rather more important	Judgements are slightly supported one over the other.
5	Much more important	Experience and Judgement strongly support one over the other.
7	Very much more important	Experience and Judgement very strongly favor one over the other.
9	Absolutely more important.	Judgements are highly favor one over the other.
2, 4, 6, 8	IntermediaryValues	Where cooperation is needed.

Analytical Hierarchy Process (AHP): The Analytical Hierarchical Processing (AHP) firstly introduced by Saaty and tells how to make a decision using AHP [19]. Initially consumers of AHP make a hierarchy of factors to decompose their decision problem. Each of them can be evaluated autonomously to build the hierarchy. Tahiri *et al.* [20], determine different factors by comparing them according to their impact on the factors in the hierarchy using AHP. AHP is used to quantify the significance of the factors. There are 25 professionals involved in judgement to formalize the significance of every factor. Every judgement are based on a scale and assigned a number. A common scale by Saaty is shown in Table 2.

The Steps to Carry out the Ahp Process Are as Follows:

Reciprocal Matrix: First, a pairwise comparison matrix has been constructed based on the factors. Every factor needs to compare with the immediate next factor [20].

Eigen Vector: Next, we have to evaluate the relative weights of the factors, which are relevant to the problem is called an eigenvector [21].

Consistency Index: Now, we have to evaluate Consistency Index (CI) for that matrix using (λmax-n)/(n-1) [21].

Consistency Ratio: Finally, we have to evaluate consistency ratio (CR) using CI divided by average consistency index (RI).

A necessary hypothesis, if factor F_1 has much more importance than factor F_2 then its value becomes 5 and F_2 importance is less than F_1 then its value will be 1/5. All pair wise comparisons are achieved through the factors, generally not exceeding to 7. Next, we have to measurethe relative weights and importance of the factors. Finally, we have to evaluate consistency ratio (CR) to see the consistency of the judgements. If the value of CR > 0.1, then judgements are unreliable.

Suppose for n number of factors, F_1, F_2….F_n are considered, which are to be compared. Relative weight of F_i relating to F_j denoted as m_{ij} and a square matrix A = [m_{ij}] of order n will be formed as given in equation (1.1).

$$A = \begin{matrix} F_1 \\ \left[m_{ij} \right] \\ F_n \end{matrix} = \begin{pmatrix} F_1 & F_1 & . & F_1 \\ 1 & m_{12} & . & m_{1n} \\ F_2 1/m_{12} & 1 & . & m_{2n} \\ . & . & . & . \\ 1/m_{1n} & 1/m_{2n} & . & 1 \end{pmatrix}$$

Here, $m_{ij} = 1/m_{ji}$ and i does not equal to j and $m_{ii} = 1$ for all i. Hence the calculated matrix is known as reciprocal matrix.

Human judgements are inconsistent. In this situation, find eigenvector ω satisfying (1.2) as:

$$A\omega = \lambda_{\max}\omega, \text{ and } \lambda_{\max} \geq n \tag{1.2}$$

Where, ω is eigen vector and $\lambda_{max}$is eigen value. The difference between eigen value and n is the inconsistency of the judgements. Saaty [19], suggested a Consistency Ratio (CR) and Consistency Index (CI) and evaluated as:

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{1.3}$$

$$CR\frac{CI}{RI} \tag{1.4}$$

Table 3: Fuzzy values used for making qualitative assessments

Fuzzy Values	Triangular Membership Function
1	(1, 1, 3)
X	(x-2, x, x+2) for x = 3, 5, 7
9	(7, 9, 11)

Where, RI is the average consistency index. Saaty proposed that if the CR > 0.1, the judgements may not be consistent and unreliable. In this situation, a new comparison matrix is needed to set up until CR < 0.1. This way we can apply the AHP for predicting a decision based on available choices at hand.

A Fuzzy AHP Multicriteria Approach for Program Selection [22]: In [9], Laarhoven *et al.* suggested the Fuzzy AHP in 1983, which was an application of the mixture of Analytic Hierarchy Process (AHP) and Fuzzy set theory. It converts the judgements of professionals from earlier specific values to fuzzy numbers and membership functions, triangular fuzzy numbers used for pair wise comparison of matrices to construct Fuzzy AHP. In this section, applicability of FAHP is discussed in detail.

Basic Methodology of FAHP: We have taken triangular fuzzy numbers for the evaluation. Fuzzy values are easily used in articulating the judgements based on the assessment of quality for testability. For the pair wise comparisons, triangular fuzzy numbers defined in Table 3 are used.

The multicriteria decision problem generally consists of following; (a) a number of alternative i.e. A_i (i = 1, 2,..., n), (b) assessment on set of criteria C_j (j = 1, 2,..., m), (c) a quantitative or qualitative assessment x_{ij} (i = 1, 2,..., n; j = 1, 2,..., m) represent every alternative A_i relating to every criterion C_j, used to make the decision matrix for alternatives, last (d) a weight vector W = (w_1, w_2,...,w_m) i.e. weights of criteria represent the significance for criteriaevaluation relating to the whole problems in achieving main objective. Steps required for the applied algorithm are as follows:

Step 1: Prepare the problem like an multi attribute problem for identifying the hierarchical structure.

Step 2: Establish the decision matrix (1.8) using (1.5) to (1.7) and Table 1.

$$C_j \text{ or } W = \begin{bmatrix} a_{11} & a_{12} & . & a_{1k} \\ a_{21} & a_{22} & . & a_{2k} \\ . & . & . & . \\ a_{k1} & a_{k2} & . & a_{kk} \end{bmatrix}, \tag{1.5}$$

Where,

$$a_{ls} = \begin{cases} 1,3,5,7,9, & l < s \\ 1, & l = s \\ \frac{1}{a_{sl}}, & l > s \end{cases} \quad (1.6)$$

$$X_{ij} \text{ or } w_j = \frac{\sum_{s=1}^{k} a_{ls}}{\sum_{l=1}^{k}\sum_{s=1}^{k} a_{ls}}, \quad (1.7)$$

$$X = \begin{bmatrix} a_{11} & a_{12} & . & a_{1m} \\ a_{21} & a_{22} & . & a_{2m} \\ . & . & . & . \\ a_{n1} & a_{n2} & . & a_{nm} \end{bmatrix} \quad (1.8)$$

Step 3: Find a weight vector (1.9) for each criteria using (1.5) to (1.7).

$$W = (w_1, w_{2...}w_m), \quad (1.9)$$

Step 4: Establish the fuzzy performance matrix (1.10) with the help of (1.8) multiplied by (1.9).

$$Z = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & . & w_j r_{1j} & \cdots & w_n r_{1n} \\ w_1 r_{i1} & w_2 r_{i2} & . & wjr_{ij} & \cdots & w_n r_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & & w_j r_{mj} & \cdots & w_n r_{mn} \end{bmatrix} \quad (1.10)$$

Step 5: Find the interval performance matrix (1.11) with the help of an α-cut on (1.10) obtained in Step 4.

$$Z_\alpha = \begin{bmatrix} [Z_{11l}^\alpha, Z_{11r}^\alpha] & [Z_{12l}^\alpha, Z_{12r}^\alpha] & \cdots & [Z_{1ml}^\alpha, Z_{1mr}^\alpha] \\ [Z_{21l}^\alpha, Z_{21r}^\alpha] & [Z_{22l}^\alpha, Z_{22r}^\alpha] & \cdots & [Z_{2ml}^\alpha, Z_{2mr}^\alpha] \\ \cdots & \cdots & \cdots & \cdots \\ [Z_{n1l}^\alpha, Z_{n1r}^\alpha] & [Z_{n2l}^\alpha, Z_{n2r}^\alpha] & \cdots & [Z_{nml}^\alpha, Z_{nmr}^\alpha] \end{bmatrix}, \quad (1.11)$$

Step 6: Obtain the crisp performance matrix (1.12) by including the decision maker's mind-set in the possibility represented by confidence index k.

$$Z_{ij\alpha}^{\lambda'} = \lambda Z_{ijr}^{\alpha} + (1-\lambda) z_{ijl}^{\alpha}, \lambda \in [0,1]$$

$$\begin{bmatrix} Z_{11\alpha}^{\lambda'} & Z_{12\alpha}^{\lambda'} & . & Z_{1m\alpha}^{\lambda'} \\ Z_{21\alpha}^{\lambda'} & Z_{22\alpha}^{\lambda'} & . & Z_{2m\alpha}^{\lambda'} \\ . & . & . & . \\ Z_{n1\alpha}^{\lambda'} & Z_{n2\alpha}^{\lambda'} & . & Z_{nm\alpha}^{\lambda'} \end{bmatrix}, \quad (1.12)$$

Step 7: Evaluate the normalized performance matrix (1.14) using (1.13).

$$Z_{ij\alpha}^{\lambda} = \frac{Z_{ij\alpha}^{\lambda'}}{\sqrt{\sum_{i=1}^{n} (Z_{ij\alpha}^{\lambda'})^2}} \quad (1.13)$$

$$\begin{bmatrix} Z_{11\alpha}^{\lambda} & Z_{12\alpha}^{\lambda} & . & Z_{1m\alpha}^{\lambda} \\ Z_{21\alpha}^{\lambda} & Z_{22\alpha}^{\lambda} & . & Z_{2m\alpha}^{\lambda} \\ . & . & . & . \\ Z_{n1\alpha}^{\lambda} & Z_{n2\alpha}^{\lambda} & . & Z_{nm\alpha}^{\lambda} \end{bmatrix} \quad (1.14)$$

Step 8: Obtain the positive ideal result as well as the negative ideal result using (1.15) and (1.16).

$$\begin{aligned} A_\alpha^{\lambda+} &= (Z_{1\alpha}^{\lambda+}, Z_{2\alpha}^{\lambda+} \cdots\cdots Z_{m\alpha}^{\lambda+}), \\ A_\alpha^{\lambda-} &= (Z_{1\alpha}^{\lambda-}, Z_{2\alpha}^{\lambda-} \cdots\cdots Z_{m\alpha}^{\lambda-}), \end{aligned} \quad (1.15)$$

Where,

$$\begin{aligned} Z_\alpha^{\lambda+} &= \max(Z_{1j\alpha}^{\lambda}, Z_{2j\alpha}^{\lambda} \cdots\cdots Z_{nj\alpha}^{\lambda}), \\ Z_\alpha^{\lambda-} &= \min(Z_{1j\alpha}^{\lambda}, Z_{2j\alpha}^{\lambda} \cdots\cdots Z_{nj\alpha}^{\lambda}), \end{aligned} \quad (1.16)$$

Step 9: Compute the level of likeness among every alternative and the positive ideal result as well as the negative ideal result by (1.17) and (1.18).

$$S_{i\alpha}^{\lambda+} = \frac{A_{i\alpha}^{\lambda} A_{i\alpha}^{\lambda+}}{\max(A_{i\alpha}^{\lambda} A_{i\alpha}^{\lambda}, A_{i\alpha}^{\lambda+} A_{i\alpha}^{\lambda+})}, \quad (1.17)$$

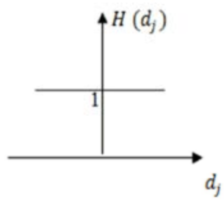
$$S_{i\alpha}^{\lambda-} = \frac{A_{i\alpha}^{\lambda} A_{i\alpha}^{\lambda-}}{\max(A_{i\alpha}^{\lambda} A_{i\alpha}^{\lambda}, A_{i\alpha}^{\lambda-} A_{i\alpha}^{\lambda-})}, \quad (1.18)$$

Step 10: Obtain the whole performance index for every alternative using (1.19):

$$P_\alpha^{\lambda} = \frac{S_{i\alpha}^{\lambda+}}{S_{i\alpha}^{\lambda+} + S_{i\alpha}^{\lambda-}}, \quad (1.19)$$

Step 11: Order the alternatives using their equivalent performance index values in the downward order.

A Preference Ranking Organization Method of Enrichment Evaluations (PROMETHEE-2) [23]: PROMETHEE-2 is a MCDM process for ranking. A preference function P_j (a, b) used in this process which depends on the pair wise comparison d_j within the computations f_j (a) and f_j (b) of alternatives a and b for criteria j. We have used usual criterion for their simplicity for our analysis as shown in below Figure 3:



$$H(d_j)=\begin{bmatrix} 0 & if & d_j \le 0 \\ 1 & if & d_j > 0 \end{bmatrix}$$

Fig. 3: Usual Criteria

Here H (d_j) represents preference function p_j (a, b). If d_j has positive number then it is represented as 1 otherwise 0. Mathematical analysis of preference function and its relationship with d_j for usual criteria function is shown in above Figure 3.

Multicriteria preference index π (a, b) and P_j (a, b) as an average weight preference function for criteria is shown below:

$$\pi(a,b)=\frac{\sum_{j=1}^{j} w_j P_j(a,b)}{\sum_{j=1}^{j} w_j} \tag{1.20}$$

$$\phi^{+}(a)=\frac{\sum_{A}\pi(a,b)}{(N-1)} \tag{1.21}$$

$$\phi^{-}(a)=\frac{\sum_{A}\pi(b,a)}{(N-1)} \tag{1.22}$$

$$\phi(a)=\phi^{+}(a)-\phi^{-}(a) \tag{1.23}$$

Where, w_j is weight of the criteria j; φ^+ (a) is the outranking index for N set of alternatives; φ^- (a) = Outranked index of a in the alternatives set N; φ (a) is the net rankingfor N set of alternatives; j = number of criteria. The higher value of φ (a) is an indicator of the best suitable alternative.

Evaluation and Analysis of Aspectj Programs Using Mcdm Approaches: In previous section, we have discussed all three popular approaches of MCDM i.e. AHP, FAHP and PROMETHEE-2 respectively. In this section, we will apply these approaches on AspectJ programs to select the best suitable choice in order to estimate their testability.

Assesment of Aspectj Programs Using AHP: Firstly, AHP is applied on the software testability relationship model shown in Figure 2 for Aspect Oriented Software.

AHP for Maintainability of Aspectj Programs: According to figure 2, we will first calculate the importance of each factor related to maintainability using AHP, thereafter testability has been computed.

Allocating the Weights to Factors: In order to assign weights to factors of software maintainability i.e. Analyzability (A_1), Changeability (A_2), Testability (A_3), Stability (A_4) and Modularity (A_5), a survey has conducted on 25 professional from academia, industry and research scholars. Considered experts are those, who have either good knowledge of OO and AO technology or are doing their research in AO development. Survey form was provided to all the experts. The survey form consist a square matrix for filling pair-wise relative weight values of factors. Firstly, square matrix is to fill pair-wise relative weight values of these five factors A_1 to A_5. The mean of collected samples of pair-wise relative weights are given in square matrix A = $[m_{ij}]$ given in equation (1.24), which is prepared using equation (1.1) to apply AHP process.

Next step is to determine Eigen vector and Eigen values to get corresponding weights of A_1, A_2, A_3, A_4 and A_5 and consistency ratio (CR).

$$A = m_{ij} = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{array} \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_5 \end{array} \\ \begin{pmatrix} 1 & 0.658 & 0.219 & 0.35 & 0.795 \\ 1.520 & 1 & 0.71 & 0.833 & 0.65 \\ 4.566 & 1.408 & 1 & 2.383 & 1.733 \\ 2.857 & 1.20 & 0.420 & 1 & 1.416 \\ 1.258 & 1.538 & 0.577 & 0.706 & 1 \end{pmatrix} \end{array} \tag{1.24}$$

Determining Eigen Vector and Eigen Values: Eigen vector can be calculated by multiplying all the values in every row of the matrix A. The n^{th} root (in our case 5^{th} root) of the product helps in getting excellent estimation to the acceptable result. The values are given in Table 4. The summation of n^{th} root is used to normalize the eigen vector and the addition should be 1.0. The 5^{th} root of first row is 0.526 and this is divided by 5.458 gives 0.096 as the first value in the eigen vector shown in Table 4. Thus Eigen vector of the relative importance of A_1, A_2, A_3, A_4 and A_5 are 0.096, 0.165, 0.353, 0.211 and 0.175, respectively, which are given in Table 4. These values are weights of main factors i.e. Analyzability (0.096), Changeability (0.165), Testability (0.353), Stability (0.211) and Modularity (0.175). Here based on the estimated values for attributes, it is clearly shown that A_3 i.e. Testability is the most important factor in software

Table 4: Eigen Vector and Eigen Value for main Factors

	A_1	A_2	A_3	A_4	A_5	n^{th} (5^{th}) root of product of	Eigen Vector (ω)
A_1	1	0.658	0.219	0.35	0.795	0.526	0.096
A_2	1.520	1	0.71	0.833	0.65	0.898	0.165
A_3	4.566	1.408	1	2.383	1.733	1.927	0.353
A_4	2.857	1.20	0.420	1	1.416	1.153	0.211
A_5	1.258	1.538	0.577	0.706	1	0.954	0.175
Total						5.458	1.000

Table 5: Satty Scale [16, 23]

1	2	3	4	5	6	7	8	9	10
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Now, one can evaluate CI using equation (1.3).

Maintainability. In order to calculate A.ω, first multiply the matrix (A_1 to A_5) from eigen vector (ω). First row of the matrix is calculated as: 1 X 0.096+0.658 X 0.165+0.219 X 0.353+0.35 X 0.211+0.795 X 0.175 = 0.495 and remaining four rows gives 0.851, 1.830, 1.079 and 0.902 respectively. As given in equation (1.2), $A.\omega = \lambda_{max}.\omega$ and $\lambda_{max} \geq 5$, next step is to get product A.ω. Eigen values λ_{max} can be evaluated by applying $\lambda_{max} = (A.\omega / \omega)$ and these five λ_{max} values are calculated as 5.156, 5.158, 5.184, 5.114 and 5.154. All these values of λ_{max} are greater than 5, which satisfy the condition of $\lambda_{max} \geq n$. The mean of λ_{max} values is 5.153.

$$CI = \frac{\lambda \max - \mathbf{n}}{\mathbf{n} - \mathbf{1}} = \frac{\mathbf{5.153 - 5}}{\mathbf{5 - 1}} = \mathbf{0.038}$$

Finally, we have to calculate CR using the CI divided by RI, the RI value is taken from Table 5 below, described by Saaty[8,19], where in upper row is the order of matrix and the lower row is the equivalent index of consistency for arbitrary judgements.

CR=CI / RI = 0.038 / 1.1 = 0.034

The calculated value of CR comes out <0.1, which means the estimate is acceptable. Hence A_3 i.e. testability has more importance than other factors of maintainability because ω shows A_3 value as 0.353 which is more than the other factors of maintainability that's why we have further classified testability and apply AHP to it to validate testability as well as its factors.

AHP for Testability of Aspectj Programs: In order to evaluate software testability using AHP we have considered five factors based on the literature review i.e. Controllability (F_1), Observability (F), $_2$Built in test Capability (F_3), Understandability (F_4) and Complexity (F_5).

Allocating the Weights to Factors: In order to assign weights to factors of software testability i.e. Controllability (F_1), Observability (F_2), Built in test Capability (F_3), Understandability (F_4) and Complexity (F_5), a survey has conducted on above considered 25 experts. The survey form consist a square matrix for filling pair-wise relative weight values of factors. Firstly, square matrix is to fill pair-wise relative weight values of these five factors F_1 to F_5. The mean of collected samples of pair-wise relative weights are given in square matrix $A = [m_{ij}]$, which is prepared using equation (1.1) to apply AHP process. Next step is to determine Eigen vector and Eigen values to get corresponding weights of F_1, F_2, F_3, F_4 and F_5 and consistency ratio (CR).

$$A = m_{ij} = \begin{array}{c} \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{array} \begin{array}{c} \begin{array}{ccccc} F_1 & F_2 & F_3 & F_4 & F_5 \end{array} \\ \begin{pmatrix} 1 & 0.983 & 0.703 & 0.477 & 0.583 \\ 1.017 & 1 & 0.828 & 0.635 & 1.708 \\ 1.422 & 1.208 & 1 & 0.815 & 2.017 \\ 2.096 & 1.575 & 1.227 & 1 & 3.183 \\ 1.715 & 0.585 & 0.496 & 0.314 & 1 \end{pmatrix} \end{array} \quad (1.25)$$

Determining Eigen Vector and Eigen Values: Several methods are there for finding the eigen vector. One of the method is by multiplying all the values in every row of the considered matrix. The n^{th} root (in our case 5^{th} root) of the product helps in getting the estimation to the acceptable result. The calculated values are given in Table 6. The sum of n^{th} root is used to find the eigen vector of factors and addition should be 1.0. In Table 6, the 5^{th} root of first row is 0.719 and this is divided by 5.29 gives 0.136 for the first factor in the eigen vector. Thus Eigen vector of the relative importance of F_1, F_2, F_3, F_4 and F_5 are 0.136, 0.186, 0.233, 0.315 and 0.130, respectively, which are given in Table 6. These values are weights of main factors i.e. controllability (0.136), observability (0.186), built in test capability (0.233), understandability (0.315) and Complexity (0.130). It clearly shown that F_4 i.e. understandability is the most important factor in context to testability. In order to calculate A.ω, first multiply the matrix (F_1 to F_5) from eigen vector (ω). First row of the matrix can be calculated as: 1 X 0.136 +0.983 X 0.186 +0.703 X 0.233 + 0.477 X 0.315 + 0.583 X 0.130 = 0.709 remaining four rows give 0.939, 1.170, 1.593 and 0.687 respectively. As given in equation (1.2), $A.\omega = \lambda_{max}.\omega$ and $\lambda_{max} \geq 5$, next step is to get product A.ω. Eigen values λ_{max} can be evaluated by applying $\lambda_{max} = (A.\omega / \omega)$ and these five λ_{max} values are calculated as 5.213, 5.048, 5.021, 5.057,

Table 6: Eigen Vector and Eigen Value for main Factors

	A_1	A_2	A_3	A_4	A_5	n^{th} (5^{th}) root of product of	Eigen Vector (ω)
F_1	1	0.983	0.703	0.477	0.583	0.719	0.136
F_2	1.017	1	0.828	0.635	1.708	0.982	0.186
F_3	1.422	1.208	1	0.815	2.017	1.231	0.233
F_4	2.096	1.575	1.227	1	3.183	1.668	0.315
F_5	1.715	0.585	0.496	0.314	1	0.690	0.130
Total						5.29	1.000

Now, one can evaluate CI using equation (1.3).

5.285. All these values of λ_{max} are greater than 5, which satisfy the condition of $\lambda_{max} \geq n$. The mean of λ_{max} values is 5.125.

Now, one can evaluate CI using equation (1.3).

$$CI = \frac{\lambda max - n}{n-1} = \frac{5.125-5}{5-1} = 0.031$$

Finally, we have to calculate CRusing the CI divided by RI, the RI value is taken from Table 5 below, described by Saaty[8,19], wherein upper row is the order of matrix and the lower row is the equivalent index of consistency for arbitrary judgements. Now, one can evaluate CR using equation (1.4).

CR=CI / RI = 0.031 / 1.1 = 0.028

The calculated CR value should be <0.1, which indicates that the estimate is acceptable. Hence the calculated value of testability is now valid. For the evaluation of quality of AO programs, we have applied AHP process on pairwise relative weights of factors from F_1 to F_5 one by one and Eigen Vectors of set of factors from Table 6 i.e. 0.136, 0.186, 0.233, 0.315 and 0.130 respectively, totaling to 1. Now, solving the pairwise comparison matrix of AO programs for each factor is shown in Table 7 to 11:

Table 7: Pairwise comparison matrix of AO Programs for Controllability

	P_1	P_2	P_3	P_4
P_1	1	3	5	7
P_2	1/3	1	3	5
P_3	1/5	1/3	1	3
P_4	1/7	1/5	1/3	1

Table 8: Pairwise comparison matrix of AO Programs for Observability

	P_1	P_2	P_3	P_4
P_1	1	3	5	7
P_2	1/3	1	3	5
P_3	1/5	1/3	1	3
P_4	1/7	1/5	1/3	1

Table 9: Pairwise comparison matrix of AO Programs for Built in Test Capability

	P_1	P_2	P_3	P_4
P_1	1	3	3	5
P_2	1/3	1	3	5
P_3	1/3	1/3	1	5
P_4	1/5	1/5	1/5	1

Table 10: Pairwise comparison matrix of AO Programs for Understandability

	P_1	P_2	P_3	P_4
P_1	1	3	3	5
P_2	1/3	1	3	5
P_3	1/3	1/3	1	5
P4	1/5	1/5	1/5	1

Table 11: Pairwise comparison matrix of AO Programs for Complexity

	P_1	P_2	P_3	P_4
P_1	1	1	1	3
P_2	1	1	1	3
P_3	1	1	1	3
P_4	1/3	1/3	1/3	1

From the above Table 7 Eigen vectors are (0.56, 0.26, 0.12 and 0.06).

From the above Table 8 Eigen vectors are (0.57, 0.26, 0.12 and 0.05).

From the above Table 9 Eigen vectors are (0.51, 0.29, 0.14 and 0.06).

From the above Table 10 Eigen vectors are (0.51, 0.29, 0.14 and 0.06).

From the above Table 11 Eigen vectors are (0.3, 0.3, 0.3 and 0.1).

Now we have to calculate overall utility of AO Programs. The summation of the products of the weight of AO Programs with reference to each factor by the weights of corresponding factor yields the global utility of each AO Programs. For example, overall utility of P_1 is computed as:

$$\text{AO Program Quality} = \sum_{i=1}^{n} \text{Comparative value of } P_i *$$

Weight value of F_i.

U (P_1) = $0.14 \times 0.56 + 0.19 \times 0.57 + 0.23 \times 0.51 + 0.31 \times 0.51 + 0.31 \times 0.3 = 0.116$

Similarly, overall utility values for other AO Programs are computed. The best AO Program is the one which is having the highest overall utility values. Accordingly, ranking of AO Programs is done which are shown in Table 12 and P1 found to be the best choice.

Table 12: Global overall utility and Rank of AO Programs

Factors	F_1	F_2	F_3	F_4	F_5	Global overall utility	Rank
Weights	0.14	0.19	0.23	0.31	0.13		
P_1	0.56	0.57	0.51	0.51	0.3	0.116	1
P_2	0.26	0.26	0.29	0.29	0.3	0.073	2
P_3	0.12	0.12	0.14	0.14	0.3	0.055	3
P_4	0.06	0.05	0.06	0.06	0.1	0.015	4

A Fuzzy AHP Multicriteria Approach for Ao Programs Selection: In this section four AO programs are considered and FAHP is applied for selection. Selecting the best program from available programs for an AO system is a difficult decision making procedure where in the whole performance of those programs desires to be estimated concerning multicriteria approach. In [24], Nagpal *et al.* proposed ANFIS scheme to measure the performance of institutional websites in similar way as we are focusing on measuring the aspect programs. However they have applied ANFIS we are using MCDM approaches. The details of programs are shown in Table 1 with fuzzy values used in Table 3. A fuzzy reciprocal judgement matrix (from F_1 to F_5) derived from the pairwise comparisons with the help the fuzzy values defined in Table 7 to 11. Here, the calculations shown are only for medium values similarly low and high values are calculated as medium values using Table 3. The steps are as follows:

Step 1: AO programs selection problem using Multicriteria approach and the hierarchical structure is shown in Figure 1.

Step 2: By relating (1.5) to (1.7) in reciprocal matrix, the rankings (x_{ij}) for the alternatives P_i (i = 1, 2, 3, 4) relating to every criterion F_j (j = 1, 2, 3, 4, 5) were calculated as:

Summation of the middle whole matrix for F_1 i.e. in Table 7 is:

$$(1+3+5+7+3^{-1}+1+3+5+5^{-1}+3^{-1}+1+3+7^{-1}+5^{-1}+3^{-1}+1)=31.53$$

$$X_1=\left[\frac{1+3+5+7}{31.53}\quad\frac{3^{-1}+1+3+5}{31.53}\quad\frac{5^{-1}+3^{-1}+1+3}{31.53}\quad\frac{7^{-1}+5^{-1}+3^{-1}+1}{31.53}\right]$$

Summation of the middle whole matrix for F_2 i.e. in Table 8 is:

$$(1+3+5+7+3^{-1}+1+3+5+5^{-1}+3^{-1}+1+3+7^{-1}+5^{-1}+3^{-1}+1)=31.53$$

$$X_2=\left[\frac{1+3+5+7}{31.53}\quad\frac{3^{-1}+1+3+5}{31.53}\quad\frac{5^{-1}+3^{-1}+1+3}{31.53}\quad\frac{7^{-1}+5^{-1}+3^{-1}+1}{31.53}\right]$$

Summation of the middle whole matrix for F_3 i.e. in Table 9 is:

$$(1+3+3+5+3^{-1}+1+3+5+3^{-1}+3^{-1}+1+5+5^{-1}+5^{-1}+5^{-1}+1)=29.59$$

$$X_3=\left[\frac{1+3+3+5}{29.59}\quad\frac{3^{-1}+1+3+5}{29.59}\quad\frac{3^{-1}+3^{-1}+1+5}{29.59}\quad\frac{5^{-1}+5^{-1}+5^{-1}+1}{29.59}\right]$$

Summation of the middle whole matrix for F_4 i.e. in Table 10 is:

$$(1+3+3+5+3^{-1}+1+3+5+3^{-1}+3^{-1}+1+5+5^{-1}+5^{-1}+5^{-1}+1)=29.59$$

$$X_4=\left[\frac{1+3+3+5}{29.59}\quad\frac{3^{-1}+1+3+5}{29.59}\quad\frac{3^{-1}+3^{-1}+1+5}{29.59}\quad\frac{5^{-1}+5^{-1}+5^{-1}+1}{29.59}\right]\backslash$$

Summation of the middle whole matrix for F_5 i.e. in Table 11 as:

$$(1+1+1+3+1+1+1+3+1+1+1+3+3^{-1}+3^{-1}+3^{-1}+1)=19.99$$

$$X_5=\left[\frac{1+1+1+3}{19.99}\quad\frac{1+1+1+3}{19.99}\quad\frac{1+1+1+3}{19.99}\quad\frac{3^{-1}+3^{-1}+3^{-1}+1}{19.99}\right]$$

Where,

$X_1 = (x_{11},x_{21},x_{31},x_{41})$, $X_2 = (x_{12},x_{22},x_{32},x_{42})$,
$X_3 = (x_{13},x_{23},x_{33},x_{43})$, $X_4 = (x_{14},x_{24},x_{34},x_{44})$,
$X_5 = (x_{15},x_{25},x_{35},x_{45})$.

Same calculations are also done for low and high values. The resulting decision matrix of an AO programs can be evaluated by fuzzy arithmetic is:

$$X=\begin{bmatrix} 0.19,0.51,1.26 & 0.19,0.51,1.26 & 0.12,0.41,1.18 & 0.12,0.41,1.18 & 0.08,0.30,1.03 \\ 0.10,0.30,0.84 & 0.10,0.30,0.84 & 0.10,0.32,0.94 & 0.10,0.32,0.94 & 0.08,0.30,1.03 \\ 0.04,0.14,0.49 & 0.04,0.14,0.49 & 0.09,0.23,0.71 & 0.09,0.23,0.71 & 0.08,0.30,1.03 \\ 0.03,0.05,0.24 & 0.03,0.05,0.24 & 0.03,0.05,0.23 & 0.03,0.05,0.23 & 0.03,0.10,0.44 \end{bmatrix}$$

Step 3: A fuzzy pairwise comparison to calculate the relative importance for selection criterion, ensuing in fuzzy reciprocal matrix (W) in below Table 13:

In the above Table 13, CR = 0.086 which is <0.1 that means pairwise comparison is acceptable. On the basis of Table 13 i.e. medium values, a low and high values matrix can be constructed using Table 3.

Summation of the middle whole matrix which is shown in Table 13 is:

$$(1+3^{-1}+3^{-1}+5^{-1}+3^{-1}+3+1+3+3^{-1}+3+3+3^{-1}+1+5^{-1}$$
$$+3^{-1}+5+3+5+1+5+3+3^{-1}+3+5^{-1}+1)=43.91$$

Calculations for the medium values of w_1, w_2, w_3, w_4, w_5:

$$\left[\frac{1+3^{-1}+3^{-1}+5^{-1}+3^{-1}}{43.91}\quad\frac{3+1+3+3^{-1}+3}{43.91}\quad\frac{3+3^{-1}+1+5^{-1}+3^{-1}}{43.91}\quad\frac{5+3+5+1+5}{43.91}\quad\frac{3+3^{-1}+3+5^{-1}+1}{43.91}\right]$$

Table 13: Pairwise comparison based on the criterion

	F_1	F_2	F_3	F_4	F_5	Eigen Vector (ω)
F_1	1	1/3	1/3	1/5	1/3	0.06
F_2	3	1	3	1/3	3	0.23
F_3	3	1/3	1	1/5	1/3	0.09
F_4	5	3	5	1	5	0.48
F_5	3	1/3	3	1/5	1	0.14
Total						1.000

Table 14: Interval performance matrix (Za)

	F1	F2	F3	F4	F5
P1	0.02,0.19	0.07,0.59	0.03, 0.29	0.1,0.84	0.03,0.35
P2	0.01, 0.13	0.04, 0.39	0.02,0.23	0.08,0.67	0.03,0.35
P3	0.004,0.07	0.02, 0.22	0.02, 0.18	0.06, 0.5	0.03,0.35
P4	0.002,0.04	0.006,0.11	0.004,0.05	0.01,0.16	0.006,0.15

Same calculations are also for low and high values. Final values for W are shown below:

$$W = \begin{bmatrix} 0.02 & 0.05 & 0.28 \\ 0.05 & 0.24 & 0.83 \\ 0.03 & 0.11 & 0.45 \\ 0.14 & 0.43 & 1.27 \\ 0.04 & 0.17 & 0.63 \end{bmatrix}$$

Step 4: A fuzzy performance matrix calculated using (1.10) as:

$$Z = \begin{bmatrix} 0.004,0.03,0.35 & 0.01,0.12,1.05 & 0.004,0.05,0.53 & 0.02,0.18,1.50 & 0.003,0.05,0.65 \\ 0.002,0.02,0.24 & 0.005,0.07,0.70 & 0.003,0.04,0.42 & 0.01,0.14,1.19 & 0.003,0.05,0.65 \\ 0.0008,0.007,0.14 & 0.002,0.03,0.41 & 0.003,0.03,0.32 & 0.01,0.10,0.90 & 0.003,0.05,0.65 \\ 0.0006,0.003,0.07 & 0.002,0.01,0.20 & 0.0009,0.006,0.10 & 0.001,0.02,0.29 & 0.001,0.01,0.28 \end{bmatrix}$$

Step 5: As an average decision α and λ taken as 0.5. Evaluate the interval performance matrix (Z_α) as in equation (1.11) by means of α-cut in the performance matrix given in Table 14:

Step 6: Construct the crisp performance matrix ($Z_\alpha^{\lambda'}$) as inequation (1.12) by including the decision makers mind-set in the possibility represented by confidence index k shown in Table 14:

$$Z_\alpha^{\lambda'} = \begin{bmatrix} 0.11 & 0.33 & 0.16 & 0.47 & 0.19 \\ 0.07 & 0.22 & 0.13 & 0.38 & 0.19 \\ 0.04 & 0.12 & 0.1 & 0.28 & 0.19 \\ 0.02 & 0.06 & 0.25 & 0.09 & 0.08 \end{bmatrix}$$

Step 7: Determine the normalized performance matrix as in equation (1.14) by equation (1.13):

$$Z_\alpha^{\lambda} = \begin{bmatrix} 0.79 & 0.79 & 0.47 & 0.70 & 0.56 \\ 0.5 & 0.52 & 0.38 & 0.58 & 0.56 \\ 0.29 & 0.29 & 0.29 & 0.42 & 0.56 \\ 0.14 & 0.14 & 0.74 & 0.13 & 0.24 \end{bmatrix}$$

Step 8: Identify the positive ideal result as well as the negative ideal result using equation (1.15) and (1.16).

$$A_\alpha^{\lambda+} = (0.79, 0.79, 0.74, 0.70, 0.24)$$

$$A_\alpha^{\lambda-} = (0.14, 0.14, 0.29, 0.13, 0, 56)$$

Step 9: Identify the extent of resemblance among every alternative and the positive ideal result as well as the negative ideal result by equation (1.17) and (1.18).

$$S_{1\alpha}^{\lambda+} = (2.66, 2.66, 2.5, 3.42, 1.56)$$

$$S_{2\alpha}^{\lambda+} = (1.60, 1.60, 0.90, 1.26, 0.35)$$

$$S_{3\alpha}^{\lambda+} = (2.34, 2.43, 1.66, 2.40, 0.79)$$

$$S_{4\alpha}^{\lambda+} = (0.37, 0.37, 1.83, 0.3, 0.19)$$

$$S_{1\alpha}^{\lambda-} = (0.41, 0.41, 0.84, 0.55, 3.14)$$

$$S_{2\alpha}^{\lambda-} = (1.1, 1.1, 1.36, 0.91, 3.14)$$

$$S_{3\alpha}^{\lambda-} = (0.7, 0.73, 1.1, 0.75, 3.14)$$

$$S_{4\alpha}^{\lambda-} = (0.44, 0.44, 4.67, 0.37, 2.91)$$

Step 10: Identify the whole performance index for every alternative by (1.19):

$$P_1 = \frac{(2.66, 2.66, 2.5, 3.42, 1.56)}{(3.07, 3.07, 3.34, 3.97, 4.7)} = (0.87, 0.87, 0.75, 0.86, 0.33)$$

$$P_2 = \frac{(1.60, 1.60, 0.90, 1.26, 0.35)}{(2.7, 2.7, 2.26, 2.17, 3.49)} = (0.59, 0.59, 0.40, 0.58, 0.10)$$

$$P_3 = \frac{(2.34, 2.43, 1.66, 2.40, 0.79)}{(3.04, 3.16, 2.76, 3.15, 3.93)} = (0.77, 0.77, 0.60, 0.76, 0.20)$$

$$P_4 = \frac{(0.37, 0.37, 1.83, 0.3, 0.19)}{(0.81, 0.81, 6.5, 0.67, 3.1)} = (0.46, 0.46, 0.28, 0.45, 0.06)$$

Step 11: Order the alternatives in the downward order by their equivalent performance index values.

The results in Table 15 show that program P_1 is the finest. Hence, it is obvious that this process can effectively imitate the imprecision and uncertainty related to the decision maker's biased judgement in human thoughts. It also presents the decision maker's as a suitable tool for better recognizes the decision problems and decision manners. Results shown in Table 15 are also similar to AHP ranking shown in Table 12.

A Preference Ranking Organization Method of Enrichment Evaluations (PROMETHEE-2) for Ao Programs Selection: In this section we have also applied the PROMETHEE-2 on same set of considered programs

Table 15: Performance Index and Ranking of the AO Programs

AO Programs	Performance Index	Ranking
P_1	0.87	1
P_2	0.59	3
P_3	0.77	2
P_4	0.46	4

Table 16: Pairwise comparison of criteria for alternatives

	F_1	F_2	F_3	F_4	F_5
P_1	0.56	0.57	0.51	0.51	0.3
P_2	0.26	0.26	0.29	0.29	0.3
P_3	0.12	0.12	0.14	0.14	0.3
P_4	0.06	0.05	0.06	0.06	0.1

of AO for cross validation purpose. First of all a pairwise comparison of programs is to be done on the basis of Table 1. Pairwise comparison of AO programs is shown from Table 7 to 11. Pairwise comparison of programs for criteria F_1, F_2, F_3, F_4 and F_5 i.e. controllability, observability, built in test capability, understandability and complexity respectively on the basis of Table 2 have done and final Eigen values are shown in Table 12.

Solution methodology of PROMETHEE-2 is divided into various steps for the computation of net φ and ranking pattern as explained below:

Step 1: Pairwise comparison of alternatives for every criteria. In this case there are five criteria for which pairwise comparison of alternatives for every criterion is to be performed. For example, for criteria F_1, pairwise difference between alternatives P_1 and P_2 are 0.56 - 0.26 = 0.3. Similarly, pairwise difference between alternatives P_1 verses P_1 is zero as they are comparing the same. Table 17 present pairwise differences between alternatives for criterion F_1.

Similarly pairwise difference matrix and preference function has been calculated for the criterion F_2, F_3, F_4 and F_5.

Step 2: Exploring preference function values for criteria F_1 to F_5 (on the basis of Figure 3 i.e. usual criteria and Table 16): Sample calculations are presented below with reference to P_1 and P_2:

Usual Criterion: Pairwise difference between alternatives P_1 and P_2 for criterion F_1 is 0.56 - 0.26 = 0.3 and corresponding preference function value P_j (P_1, P_2) is 1 (as 0.3 is positive) and similarly for P_2 to P_1 pairwise difference is -0.3 and corresponding preference function value Pj (P2, P1) is taken as 0 (as negative value). In case of usual criterion function, elements of preference function matrix are either 0 or 1.

Step 3: Computation of multicriterion preference Index based on Eq. 1.20 and π (P_1, P_2) is computed as follows:

Preference function values for P_1 and P_2 for criterion F_1 to F_5 are 1, 1, 1, 1 and 0. Corresponding weights of criteria are 0.14, 0.19, 0.23, 0.31 and 0.13 (AHP weights of factors of testability from Table 6). Substitution in Eq. 1.20 yields,

$$\phi^{+}(P_1)=\frac{0+0.87+0.87+1}{3}=0.913$$

Similarly, φ+ (P_2), φ+ (P_3), φ+ (P_4) are 0.623, 0.333 and 0 respectively.

Step 4: Computation of φ^+ (as per Eq. 1.21)

$$\phi^{+}(P_1)=\frac{0+0.87+0.87+1}{3}=0.913$$

Similarly φ^+ (P_2), φ^+ (P_3), φ^+ (P_4) are 0.623, 0.333 and 0 respectively.

Step 5: Computation of φ^- (as per Eq. 1.22)

$$\phi^{-}(P_1) = \frac{0}{3} = 0$$

Similarly φ^- (P_2), φ^- (P_3), φ^- (P_4) are 0, 0.29, 0.58 and 1 respectively.

Step 6: Computation of net φ (as per Eq. 1.23)

Net φ (P_1) = φ^+ (P_1) - φ^- (P_1) = 0.913 - 0 = 0.913

Similarly φ (P_2), φ (P_3), φ (P_4) values for other alternatives are calculated. The higher value of net φ denotes to the best alternative. Table 19 represents φ^+, φ^-, net φ and corresponding ranking pattern of each alternative.

It is observed from Table 19 that ranking pattern in the order of alternatives from P_1 to P_4, alternative P_1 with the maximum net φ value is considered as the best suitable program. Now we conclude based on the Tables 12, 15 and 19 that ranking calculated by us true and cross validated. These approaches can also help the software professionals in selecting the best suitable choices in term of software quality. We have taken testability as an attribute for selection among the available choices of AO programs and found program P1 as the best suitable choice.

Table 17: Pairwise difference matrix and preference function values for F1

Alternatives	P_1	P_2	P_3	P_4
P_1	0 (0)	0.3 (1)	0.44 (1)	0.5 (1)
P_2	-0.3 (0)	0 (0)	0.14 (1)	0.2 (1)
P_3	-0.44 (0)	-0.14 (0)	0 (0)	0.06 (1)
P_4	-0.5 (0)	-0.2 (0)	-0.06 (0)	0 (0)

Table 18: Multicriteria Preference Index Values

	P_1	P_2	P_3	P_4
P_1	0	0.87	0.87	1
P_2	0	0	0.87	1
P_3	0	0	0	1
P_4	0	0	0	0

Table 19: φ^+, φ^-, net φ and ranking pattern of each alternative

Alternatives	Φ^+	Φ^-	Net Φ	Rank
P1	0.913	0	0.913	1
P_2	0.623	0.29	0.333	2
P_3	0.333	0.58	-0.247	3
P_4	0	1	-1	4

CONCLUSION

Present work relates the analysis of four aspect oriented programs based on the factors of software testability i.e. controllability, observability, built in test capability, understandability and complexity by pairwise comparison of factors and as well as programs. The evaluation is based on the Multicriteria decision making (MCDM) approaches that are analytical hierarchy process (AHP), Fuzzy AHP and PROMETHEE-2. Firstly, the validation of testability has been made through AHP as a main sub characteristic of maintainability. Thereafter, validation of all factors of software testability has been made. The relative weight values taken through a survey based on 25 experts of the domain area and mean of the values have been taken as a sample. From the above calculations of AHP which is applied on testability, it has found that understandability is the most important factor of software testability. In fuzzy AHP and PROMETHEE-2 approaches a pairwise comparison of programs has been evaluated and ranked accordingly. From these measurements of programs, P_1 i.e. Observer program found to be best. Hence, fuzzy AHP and PORMETHEE-2 are efficiently used for testability of aspect oriented programs and choosing the best ones. These approaches can be adopted by software professionals to choose the best suitable software in term of quality attribute. In the similar way quality attributes can be quantitatively measured using MCDM approaches and help in selecting best suitable choice for use.

In future these programs results can further be cross validated using the design metrics and other approaches in term of maintainability for real life projects too. Neural networks, support vector machine, TOPSIS approaches can be considered for the evaluation of similar results from multiple dimensions. Software practitioners can use the proposed approach for selecting the appropriate program in term of software testability for AO software. In future similar approaches may be used to evaluate the other quality attributes for Aspect Oriented Software.

ACKNOWLEDGMENT

We would like to express our sincere gratitude toward the faculties of Amity University and Gautam Buddha University for helping us in refining the objectives and both universities for providing us research environment and facilities. We also like to thank Dr. Arun Sharma, Professor and Head, KIET Ghaziabad, India for his valuable suggestions and review comments for this manuscript.

REFERENCES

1. Bruntink, M. and A.V. Deursen, 2004. Predicting Class Testability using Object-Oriented Metrics, Published in proceedings of 4th IEEE International Workshop on Source Code Analysis and Manipulation, pp: 136-145, Chicago, US.
2. IEEE Standard Glossary of Software Engineering Terminology, IEEE, 1990.
3. Mary, J.H., 2000. Testing: a Roadmap, Published in proceedings of the Conference on The Future of Software Engineering, pp: 61-72, Limerick, Ireland.
4. Laarhoven, P.J.M. and W. Pedrycz, 1983. A Fuzzy Extension of Saaty's Priority Theory, Fuzzy Set Systems, 11: 229-241.
5 Stewart, T., 1992. A Critical Survey on the Status of multiple Criteria Decision Making Theory and Practice, Omega, 20: 569-586.
6. Duckstein, L., A. Tecle, H. P. Nachnebel, B. F. Hobbs, 1989. Multicriteria Analysis of Hydropower operation, Journal of Energy Engineering, 115(3): 132-153.
7. Koontz, H. and H. Weihrich, 1990. Essentials of Management, McGraw-Hill publishing company, Fifth Edition.
8. Saaty, T.L., 2008. Decision Making with the Analytic Hierarchy Process, International Journal of Services Sciences, 1(1): 83-98.

9. Singh, P.K., O.P. Sangwan, A. Pratap and A.P. Singh, 2014. An Analysis on Software Testability and Security in Context of Object and Aspect Oriented Software Development, International Journal of Security and Cybercrime, Romania, 3(1): 17-28.
10. Tomić Vojislav, Marinković Zoran and Janošević Dragoslav, 2011. PROMETHEE Method Implementation with Multi-criteria Decisions, FactaUniversitatis series: Mechanical Engineering, 9(2): 193 - 202.
11. Behzadian, M., R.B. Kazemzadeh, A. Albadvi and M. Aghdasi, 2010. PROMETHEE: A Comprehensive literature Review on Methodologies and Applications, European Journal of Operation Research, 200(1): 198-215.
12. Challa, J.S., A. Paul, Y. Dada, V. Nerella, P.R. Srivastava and A.P. Singh, 2011. Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach, Journal of Information Processing Systems, 7(3): 473-518.
13. Singh, P.K. and O.P. Sangwan, 2013. Aspect Oriented Software Metrics Based Maintainability Assessment: Framework and Model, published in proceeding's of Confluence-2013, The Next Generation Information Technology Submit, Amity University, Noida, India.
14. Kumar, A., P.S. Grover and R. Kumar, 2009. A Quantitative Evaluation of Aspect-Oriented Software Quality Model (AOSQUAMO), ACM SIGSOFT, Software Engineering Notes. 34(5): 1-9.
15. Santos, R.P., H.A.X. Costa, P.A.P. Júnior, A.F. Amâncio, A.M.P. Resende and C.M.L. Werner, 2008. An Approach Based on Maintainability Criteria for Building Aspect-Oriented Software Implementation Model, INFOCOMP Journal of Computer Science, Special Edition, pp: 11-20.
16. Rønningen, E. and T. Steinmoen, 2004. Metrics for Aspect-Oriented Programming of middleware systems, Department of Computer and Information Science (IDI), Norwegian University (NTNU), Trondheim.
17. Sharma, V. and P. Baliyan, 2011. Maintainability Analysis of Component Based Systems, International Journal of Software Engineering and Its Applications, 5(3): 107-117.
18. Ghosh, S., S.K. Dubey and A. Rana, 2011. Comparative Study of the Factors that Affect Maintainability, International Journal on Computer Science and Engineering (IJCSE), 3: 12.
19. Saaty, T.L., 1990. How to make a decision: The Analytical Hierarchy Process, European Journal of Operation Research, 48: 9-26, New Holland.
20. Tahiri, F., M.R. Osman, A. Ali and R.M. Yusuff, 2008. AHP Approach for Supplier Evaluation and Selection in a Steel Manufacturing Company, 1(2): 54-76.
21. Coyle, G., 2004. The Analytic Hierarchy Process (AHP), Practical Strategy, Open Access Material, Pearson Education.
22. Deng, H., 1999. Multicriteria Analysis with Fuzzy Pairwise Comparison, International Journal of Approximate Reasoning, 21: 215-231.
23. Raju, S.K. and N.D. Kumar, 2010. Multicriterion Analysis in Engineering and Management, PHI Publication, pp: 103-107.
24. Nagpal, R., D. Mehrotra, A. Sharma and P. Bhatia, 2013. ANFIS method for usability assessment of website of an Educational Institute, World Applied Sciences Journal, 23(11): 1489-1498.
25. Singh, P.K., O.P. Sangwan, A. Pratap and A.P. Singh, 2014. A Quantitative Evaluation of Reusability for Aspect Oriented Software using Multi-criteria Decision Making Approach World Applied Sciences Journal, 30(12): 1966-76.