# Fuzzy Layered Approach for Maintainability Evaluation of Object Oriented Software System

Neha Goel, Sanjay Kumar Dubey, Prof. (Dr.) Ajay Rana

**Abstract**— A large number of software products enter the maintenance phase due to the growing application of information systems. Software maintenance is the modification of a software product after delivery to correct faults and improve its overall performance and quality. Easily maintainable software saves large costs and effort involved in developing the software. Hence maintainability assessment is an essential component of software development life cycle. Its measurement is possible only by quantifying the sub-characteristics affecting software maintainability. This paper develops a comprehensive model to evaluate the maintainability of object-oriented software system based on fuzzy layered approach. The evaluation result is objective and provides a guideline to design a software product with high maintainability.

**Index Terms**— Maintainability, Object-oriented, Fuzzy, AHP, Model, Metrics, System

————————— ◆ —————————

## 1 INTRODUCTION

A software product requires a number of measures to be taken into account for its designing. The most important measure that must be considered in any software product is its design quality [1]. Among all the quality criteria, software maintainability is broadly accepted as a highly significant quality criterion in the economic success of engineering systems and products. There is a need for software engineers to understand how various components of a design interact in order to maintain and enhance the reliability of software during maintenance. Maintenance of software is one of the most expensive and resource requiring phase of the software development process. Statistics from various organizations shows that 40% to 80% of the development expenditure on the average software is spent in the 'maintenance' phase in which bugs are fixed, features are enhanced, and the software is updated to keep pace with changing domain requirements [2], [3]. Thus maintainability evaluation is an essential component of modern software development life cycle. Evaluation of software maintainability, if done accurately, can be useful in aiding decision making related to the software, efficiency of the maintenance process, comparing productivity and costs among different projects, allocation of resource and staff, and so on. This minimizes the future maintenance effort [4].

Assessing maintainability of a system is a difficult process as many contradictory criteria must be considered in order to reach a decision [5]. Hence a layered approach is used to evaluate software maintainability [6]. In this approach, fuzzy evaluation method in combination with Analytic Hierarchy Process (AHP) is utilized to handle problems involving multiple indices based on quantitative procedural information to get the qualita-

tive results. AHP [7] is used since it helps to capture both subjective and objective evaluation measures, providing a useful mechanism for checking the consistency of the evaluation measures and suggested alternatives thus reducing bias in decision making.

The study has been conducted in object-oriented paradigm. This is due to the fact that the primary purpose of object-oriented design is to improve software quality criteria such as maintainability, reliability, usability, etc by managing software complexity. The logical complexity of the source code has a strong correlation to the maintainability of the resultant software [8], [9]. Reducing the software development and maintenance costs is the main objective of object-oriented design. In order to facilitate the analysis and evaluation of maintainability of an object-oriented system, Chidamber and Kemerer (CK) metrics [10] have been used. CK metrics are design complexity metrics that aid in identifying certain design and code characteristics in object-oriented software which in turn helps in assessing external software qualities such as software defects, testing, and maintenance effort [11]. Hence the main objective of this paper is to evaluate software maintainability by using fuzzy layered evaluation method in combination with Analytic Hierarchy Process (AHP).

## 2 LITERATURE SURVEY

Several studies have been conducted to assess maintainability using fuzzy approach. K.K. Aggarwal et al. [12] proposed an integrated measure of software maintainability based of fuzzy theory utilizing three important aspects of software- Readability of Source Code (RSC), Documentation Quality (DOQ), and Understandability of Software (UOS). In [13] a maintainability evaluation approach based on fuzzy logic where fuzzy linguistic variables are employed in order to represent and handle the design data available early in the design process was presented. A maintainability evaluation model in virtual environment was built by analyzing the maintenance task for each

————————————————————
- *Neha Goel is with Amity University, Sec-125, NOIDA (U.P.) 201303, India (+91-9717941986. e-mail: nehank21@gmail.com).*
- *Sanjay Kumar Dubey is with Amity University, Sec-125, NOIDA (U.P.) 201303, India (e-mail: skdubey1@amity.edu).*
- *Prof. (Dr.) Ajay Rana is with Amity University, Sec-125, NOIDA (U.P.) 201303, India (e-mail: ajay_rana@amity.edu).*

replaceable unit of product in [14]. A fuzzy logic based precise and easy approach to quantify maintainability of software was developed using four major aspects of software i.e., average numbers of live variables, average life span of variables, average cyclomatic complexity and the comment ratio in [15]. A fuzzy logic based approach was presented to estimate the maintainability for component-based systems in [16]. A fuzzy index system utilizing AHP was proposed to solve the problem of software maintainability metrics in [6]. A new synthetic maintainability evaluation model based on fuzzy entropy theory was recommended in [17]. Jing Rong Li et al. [18] analyzed the tribo-maintainability related design factors from systematic perspective and proposed a fuzzy set based approach to quantitatively evaluate design for the tribo-maintainability. Qingbo Hao et al. [19] presented a new method for maintainability allocation, which comprehensively applied interval analysis, fuzzy comprehensive evaluation, and analytic hierarchy process (AHP). Haiquan Yu et al. [20] provided a fuzzy comprehensive evaluation method for product maintainability evaluation in virtual environment. A detailed practical model was proposed to evaluate the maintainability of software services in service-oriented architecture based on the fuzzy system in [21]. In [22], a system based on fuzzy inference approach which uses activity-based quality model to deal with maintainability was proposed. A lot more similar studies can be found elsewhere in literature.

## 3 SOFTWARE MAINTAINABILITY

According to IEEE standard glossary of Software Engineering, maintainability is "the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment" [23].

The ISO/IEC-9126 standard [24] describes a model for software product quality that dissects the overall notion of quality into 6 main characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. These characteristics are further subdivided into 27 sub-characteristics. Maintainability is one of the main criteria, characteristics or contributing attributes towards quality. It is the capability of the software to be modified [24]. It is characterized by the following sub-criteria:

1. Analyzability- It is the capability of software to be diagnosed for deficiencies or causes of failures in the software or for identification of parts requiring modification.
2. Changeability- It is the capability of software to enable a specified modification to be implemented.
3. Stability- It is the ability of software to minimize unexpected effects from software modifications.
4. Testability- It is the ability of software to validate modified software.

## 4 ANALYTIC HIERARCHY PROCESS (AHP)

AHP, as proposed by Saaty in 1980, is a multi-criteria decision making method for complicated and unstructured problems and it is also an approach that uses a hierarchical model having levels of goal, criteria, possible sub-criteria, and alternatives [25]. With AHP, the decision maker selects the alternative that best meets his or her decision criteria developing a numerical score to rank each decision alternative based on how well each alternative meets them. In other words, it is an approach that is suitable for dealing with complex systems where both qualitative and quantitative aspects need to be considered.

AHP was used in flexible manufacturing systems [26] and in integrated manufacturing [27]. AHP applications in banking include work on bank strategic planning focusing on merger and acquisitions process [28] and setting up development goals in low-income developing countries [29]. Use of AHP in the evaluation of technology investment decisions [30], and layout design [31] was also fundamental. AHP process has been applied to software selection in [32], [33], [34]. A model for bank performance evaluation and rating highlighting CAMEL rating [35] was based on AHP. AHP has also been utilized in enhancement of financial risk assessment [36]. Data mining along with AHP was used to evaluate a software system's maintainability according to the ISO/IEC-9126 quality standard in [37]. A fuzzy comprehensive model involving AHP and fuzzy theory for evaluating usability was proposed in [38]. AHP was used in banking crisis resolution in Indonesia [39]. A decision model based on AHP and TOPSIS technique was proposed by [40] in order to help human resources managers in bank and insurance companies in hiring more qualified graduates for their companies. AHP was used in evaluation of software by evaluators with little information technology experience in [41]. A methodology for source code quality and static behavior evaluation of a software system using AHP was proposed by [42]. Application of Excel to calculate the weights of software maintainability evaluation based on AHP was recommended in [43]. A multi attributes decision model was recommended to evaluate certain chosen solutions in the case of U.S economic crisis in [44]. AHP was used to examine and evaluate the current e-payment systems in [45]. Furthermore, many applications of AHP developed by various authors can be found in literature.

## 5 OBJECT-ORIENTED METRICS

As discussed in section 2, maintainability evaluation into object-oriented paradigm uses fuzzy layered technique [6]. In order to do so, Chidamber and Kemerer (CK) software metrics [10] have been used. These metrics are aimed at assessing the design of object-oriented system rather than implementation. This makes them more suited to object-oriented paradigm as object-oriented design puts great emphasis on the design phase of software system [46]. The CK metric suite consists of six design complexity metrics- WMC, DIT, NOC, CBO, RFC and LCOM. Except for LCOM, all these metrics can be used as maintainability predictors as LCOM is uncorrelated with the maintainability of the software [47]. The CK metrics are briefly described as follows [10]:

## 5.1 WMC (Weighted Methods per Class)

It is a weighted sum of all the methods defined in a class. It measures the complexity of a class. It also predicts how much time and effort is required to develop and maintain the class. High WMC indicates greater complexity and hence low maintainability.

## 5.2 DIT (Depth of Inheritance Tree)

It is the length of the longest path from a given class to the root class in the inheritance hierarchy and is measured by the number of ancestor classes. So this metric calculates how far down a class is declared in the inheritance hierarchy. High DIT indicates greater design complexity and more fault-proneness.

## 5.3 NOC (Number Of Children)

It is equal to the number of immediate child classes derived from a base class. High NOC means greater level of reuse, more effort required for testing, more complexity and fault-proneness.

## 5.4 CBO (Coupling Between Objects)

For a class, CBO is measured by counting the number of other classes to which it is coupled. Coupling is a measure of inter-dependence of two objects. Two classes are coupled if methods of one use methods and/or instance variables of the other. High CBO indicates complex design, decreases modularity, and complicates testing of the class.

## 5.5 RFC (Response For a Class)

It is the count of all the methods which can potentially be executed (directly or indirectly) in response to a message to an object of that class or by some method in the class. (This includes all methods accessible within class hierarchy). High RFC means more effort required for testing, greater design complexity and fault-proneness.

The values of all the above metrics are inversely proportional to the maintainability of a system [48].

## 6 PROPOSED MAINTAINABILITY MODEL

ISO/IEC 9126, 2001[24] identifies the major quality criteria of a software system. It defines a software quality model in terms of criteria, sub-criteria and metrics. Maintainability is an important quality criterion in the cycle of software product evolution. It includes four sub-criteria which are analyzability, changeability, stability and testability. This paper proposed a layered approach of Maintainability model for object-oriented software systems. This approach includes three levels, the 1st layer corresponding to the criterion, the 2nd layer corresponding to sub-criteria and the 3rd layer corresponding to metrics impacting sub-criteria. The hierarchical structure of this layered approach is shown in Fig. 1.
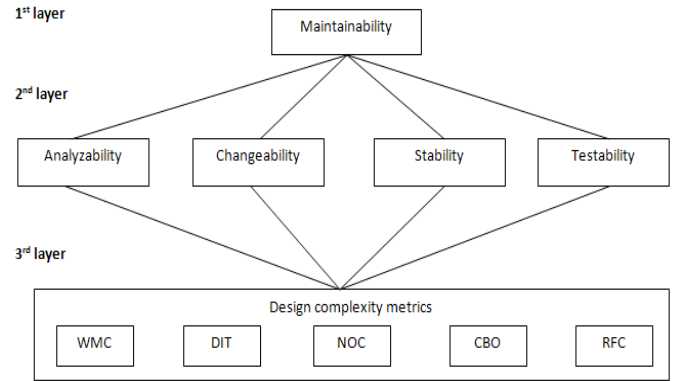


Fig. 1. Proposed Maintainability Hierarchy

The layered approach deals with program information from bottom to top which starts from simple and measurable data and analyzes the quality of software maintainability finally, as follows:

1. Compute the values of the metrics in 3rd layer using appropriate tools and determine whether each value falls within the scope of the permission as defined by NASA [49], if it is true, its value equals to 1, otherwise 0.
2. Compute the value of each sub-criterion in 2nd layer in accordance with the relationship between the 3rd and 2nd layers.
3. Evaluate the qualitative value of the 1st layer criterion, maintainability in accordance with the relationship between the 2nd and 1st layers.

We use AHP to calculate the weight vectors at each layer to the upper layers.

TABLE 1
CK METRICS

| Metric | Permissible ranges [49] |
|--------|------------------------|
| WMC | 0-20 |
| DIT | 0-3 |
| NOC | 0-1 |
| CBO | 0-5 |
| RFC | 50-100 |

In order to evaluate the qualitative value of all the layers, we perform the following steps [6]:

1. Firstly, the evaluation vector is determined in terms of certain number of grades based on the traditional fuzzy theory. According to this theory, the evaluation vector for evaluation objects should be defined as follows:
   V = (V1, V2, V3, …….., Vn)
   In our evaluation approach, the evaluation vector is:
   V = (bad, average, good, excellent)

2. Secondly, the threshold (v1, v2, v3, v4, v5) of the sub-criteria based on expert opinion is determined.
3. Thirdly, the quality of main criterion, maintainability is obtained by comparing the calculated value of the sub-criteria with the threshold.

To establish the relationship between the layers by grade, we determine membership degrees deal with the sub-criteria used for evaluation in the fuzzy method.

# 7 METHODOLOGY

The methodology adopted here involves the use of AHP along with fuzzy layered technique.

## 7.1 Analytic Hierarchy Process (AHP)

The AHP is a decision – making and estimation method which gives the percentage distribution of decision points according to factors affecting decision, that is used if there is a defined decision hierarchy [7]. Based on Fig. 1 and Table 1, AHP is applied to determine the weight vectors at all levels as follows:

1. The method of pair-wise comparisons and 1-9 scale is used to form the judgment matrix Rk between the 2nd layer sub-criteria and 3rd layer metrics. (k=1, 2, 3, 4)
2. The $n^{th}$ root (where n is the number of attributes to be compared) and the corresponding eigenvector wk(3) is calculated, after which consistency test is performed. Consistency Index (CI) is calculated as:

   CI = $(\lambda_{max} – n)/(n – 1)$
   where, $\lambda_{max}$ = the biggest eigen value of matrix $R_k$

   Consistency Ratio (CR) measures how consistent the judgments have been relative to large samples of purely random judgments. CR is calculated by dividing the Consistency Index (CI) for the set of judgments by the Index for the corresponding random matrix (RI) as given by (Thomas L. Saaty, 1980).

   If CR = CI/RI < 0.1 then Rk is correct, otherwise Rk needs to be modified.
3. The matrices are combined as w(3) = (w1(3), w2(3), w3(3), w4(3))' to obtain the final weight vector matrix between 2nd layer sub-criteria and 3rd layer metrics as,

$$w(3) = \begin{pmatrix} w11 & w12 & w13 & w14 \\ w21 & w22 & w23 & w24 \\ w31 & w32 & w33 & w34 \\ w41 & w42 & w43 & w44 \\ w51 & w52 & w53 & w54 \end{pmatrix}$$

4. The eigenvector w(2) between the 1st layer criterion and 2nd layer sub-criteria is calculated as the weight vector in the same way to obtain the matrix as,

$$w(2) = \begin{bmatrix} u11 & u12 & u13 & u14 \end{bmatrix}'$$

where $[\ ]'$ notation denotes the transpose of a matrix.

## 7.2 Fuzzy Layered Technique

This technique has been derived from [6].
The value of each 3rd layer metrics is calculated and verified whether it is within the permission of metrics [49] or not, if it is true, the value equals to 1, otherwise it is 0. Then, the value vector T of the 2nd layer sub-criteria is calculated by multiplying the matrix of values of the 3rd layer metrics (value) and the weight vector w(3),

$$T = value * w(3) = \begin{bmatrix} T11 & T12 & T13 & T14 \end{bmatrix}$$

The evaluation vector is V = (V1, V2, V3, V4) = (bad, average, good, excellent). In this paper, the semi-trapezoidal distribution and the trapezoidal distribution [50] is used in assessing the membership degrees.
The threshold of the sub-criteria obtained through the expert method is (v1, v2, v3, v4, v5).
e1, e2, e3, e4 denote the median of (v1, v2), (v2, v3), (v3, v4), (v4, v5) respectively.
Hence we can get the membership degrees in V as follows,

$$B1(v) = \begin{cases} 1 & v1 <= v <= v2 \\ (v-e2)/(v2-e2) & v2 < v < e2 \\ 0 & otherwise \end{cases} \quad (1)$$

$$B2(v) = \begin{cases} 1 & v2 <= v <= v3 \\ (v-e1)/(v2-e1) & e1 < v < v2 \\ (v-e3)/(v3-e3) & v3 < v < e3 \\ 0 & otherwise \end{cases} \quad (2)$$

$$B3(v) = \begin{cases} 1 & v3 <= v <= v4 \\ (v-e2)/(v3-e2) & e2 < v < v3 \\ (v-e4)/(v4-e4) & v4 < v < e4 \\ 0 & otherwise \end{cases} \quad (3)$$

$$B4(v) = \begin{cases} 1 & v4 <= v <= v5 \\ (v-e3)/(v4-e3) & e3 < v < v4 \\ 0 & otherwise \end{cases} \quad (4)$$

For Analyzability, by putting v = T11 in (1) - (4), we can get the membership degree as below,

B1(v) = x11, B2(v) = x21 , B3(v) = x31 , B4(v) = x41

Similarly, the fuzzy evaluation matrix of the 2nd layer sub-criteria is obtained as,

$$F(2) = \begin{pmatrix} x11 & x12 & x13 & x14 \\ x21 & x22 & x23 & x24 \\ x31 & x32 & x33 & x34 \\ x41 & x42 & x43 & x44 \end{pmatrix}$$

Thus the fuzzy evaluation matrix of the 1st layer criterion is,

$S = F(2) * w(2) = \begin{bmatrix} s1 & s2 & s3 & s4 \end{bmatrix}'$

S shows the quality of the 1st layer criterion. The membership degree in bad case is s1, the membership degree in average case is s2, the membership degree in good case is s3 and the membership degree in excellent case is s4. According to the maximal membership degree principle [51], we can infer whether the maintainability of the software product is bad, average, good or excellent.

## 8 CASE STUDY

On the basis of the fuzzy layered evaluation model proposed above, two object-oriented software products namely Y and Z will be evaluated.

### Evaluation of Software Y:

According to the AHP method, reciprocal matrices $R_k$ are formed between the 2nd layer sub-criteria and 3rd layer metrics and consistency test is performed as shown below,

**Analyzability (R1)**

TABLE 2
RECIPROCAL MATRIX FOR ANALYZABILITY

| | WMC | DIT | NOC | CBO | RFC | Eigenvector |
|------|------|------|------|------|------|------|
| WMC | 1.000 | 0.333 | 0.333 | 0.143 | 1.000 | 0.055 |
| DIT | 3.000 | 1.000 | 3.000 | 0.143 | 3.000 | 0.165 |
| NOC | 3.000 | 0.333 | 1.000 | 0.111 | 2.000 | 0.093 |
| CBO | 7.000 | 7.000 | 9.000 | 1.000 | 7.000 | 0.628 |
| RFC | 1.000 | 0.333 | 0.500 | 0.143 | 1.000 | 0.059 |
| Total | | | | | | 1.000 |

$CI = (\lambda_{max} - n)/(n - 1) = (5.313 - 5)/(5 - 1) = 0.078$
$CR = CI/RI = 0.078/1.12 = 0.069 < 0.1$

TABLE 3
RECIPROCAL MATRIX FOR CHANGEABILITY

| | WMC | DIT | NOC | CBO | RFC | Eigenvector |
|------|------|------|------|------|------|------|
| WMC | 1.000 | 0.333 | 0.333 | 0.111 | 1.000 | 0.049 |
| DIT | 3.000 | 1.000 | 5.000 | 0.143 | 5.000 | 0.192 |
| NOC | 3.000 | 0.200 | 1.000 | 0.143 | 2.000 | 0.084 |
| CBO | 9.000 | 7.000 | 7.000 | 1.000 | 9.000 | 0.626 |
| RFC | 1.000 | 0.200 | 0.500 | 0.111 | 1.000 | 0.049 |
| Total | | | | | | 1.000 |

**Changeability (R2)**

$CI = (5.395 - 5)/(5 - 1) = 0.395/4 = 0.099$
$CR = CI/RI = 0.099/1.12 = 0.088 < 0.1$

TABLE 4
RECIPROCAL MATRIX FOR STABILITY

| | WMC | DIT | NOC | CBO | RFC | Eigenvector |
|------|------|------|------|------|------|------|
| WMC | 1.000 | 1.000 | 0.333 | 0.200 | 1.000 | 0.096 |
| DIT | 1.000 | 1.000 | 1.000 | 0.333 | 1.000 | 0.132 |
| NOC | 3.000 | 1.000 | 1.000 | 0.333 | 1.000 | 0.165 |
| CBO | 5.000 | 3.000 | 3.000 | 1.000 | 5.000 | 0.487 |
| RFC | 1.000 | 1.000 | 1.000 | 0.200 | 1.000 | 0.119 |
| Total | | | | | | 0.999 |

**Stability (R3)**

$CI = (5.141 - 5)/(5 - 1) = 0.141/4 = 0.035$
$CR = CI/RI = 0.035/1.12 = 0.031 < 0.1$

TABLE 5
RECIPROCAL MATRIX FOR TESTABILITY

| | WMC | DIT | NOC | CBO | RFC | Eigenvector |
|------|------|------|------|------|------|------|
| WMC | 1.000 | 0.333 | 1.000 | 0.143 | 1.000 | 0.078 |
| DIT | 3.000 | 1.000 | 3.000 | 0.200 | 3.000 | 0.199 |
| NOC | 1.000 | 0.333 | 1.000 | 0.200 | 3.000 | 0.103 |
| CBO | 7.000 | 5.000 | 5.000 | 1.000 | 5.000 | 0.553 |
| RFC | 1.000 | 0.333 | 0.333 | 0.200 | 1.000 | 0.067 |
| Total | | | | | | 1.000 |

**Testability (R4)**

$CI = (5.279 - 5)/(5 - 1) = 0.279/4 = 0.069$
$CR = CI/RI = 0.069/1.12 = 0.062 < 0.1$

As CR< 0.1, the above comparative judgment matrices have a satisfactory consistency and the indicator weight coefficients calculated through these matrices are acceptable.

So, the matrix composed by weight vectors between the 2nd layer sub-criteria and 3rd layer metrics is,

$w(3) = \begin{bmatrix} 0.055 & 0.049 & 0.096 & 0.078 \\ 0.165 & 0.192 & 0.132 & 0.199 \\ 0.093 & 0.084 & 0.165 & 0.103 \end{bmatrix}$

|  | | | |
|---|---|---|---|
| 0.623 | 0.626 | 0.487 | 0.553 |
| 0.059 | 0.049 | 0.119 | 0.067 |

Similarly, reciprocal matrix is formed between the 1st layer criterion and 2nd layer sub-criteria and consistency test is performed as shown below,

<div align="center">

TABLE 6
RECIPROCAL MATRIX FOR MAINTAINABILITY

</div>

|  | Analyzability | Changeability | Stability | Testability | Eigenvector |
|---|---|---|---|---|---|
| Analyzability | 1 .000 | 0.200 | 0.111 | 0.333 | 0.046 |
| Changeability | 5.000 | 1.000 | 0.333 | 3.000 | 0.237 |
| Stability | 9.000 | 3.000 | 1.000 | 9.000 | 0.625 |
| Testability | 3.000 | 0.333 | 0.111 | 1.000 | 0.091 |
| Total |  |  |  |  | 0.999 |

**Maintainability**

$CI = (4.107 – 4)/(4 – 1) = 0.107/3 = 0.036$
$CR = CI/RI = 0.036/0.90 = 0.040 < 0.1$

We obtain a satisfactory consistency, and hence the indicator weight coefficients calculated through this matrix is acceptable.

So, the matrix between the 1st layer criterion and 2nd layer sub-criteria is,

$$w(2) = \begin{bmatrix} 0.046 & 0.237 & 0.625 & 0.091 \end{bmatrix}'$$

The value of each 3rd layer metric is calculated and verified

<div align="center">

TABLE 7
VALUES OF THE 3RD LAYER METRICS

</div>

| Sub-criteria | Low-High [49] | Current | Value |
|---|---|---|---|
| WMC | 0-20 | 9.6 | 1 |
| DIT | 0-3 | 1.19 | 1 |
| NOC | 0-1 | 0.54 | 1 |
| CBO | 0-5 | 9.74 | 0 |
| RFC | 50-100 | 56.3 | 1 |

whether it is within the permission of metrics [49] or not, if it is true, the value equals to 1, otherwise it is 0, as shown in Table 7.

Then, the value vector T of the 2nd layer sub-criteria is calculated with the values of the 3rd layer metrics (value) and the weight vector w(3) ,

$$T = value * w(3) = \begin{bmatrix} 0.372 & 0.374 & 0.512 & 0.447 \end{bmatrix}$$

The evaluation vector is V = (V1, V2, V3, V4) = (bad, average, good, excellent).

The threshold of the sub-criteria obtained through the expert method is,
$(v1, v2, v3, v4, v5) = (0, 0.25, 0.5, 0.75, 1)$

e1, e2, e3, e4 which denote the median of (v1, v2), (v2, v3), (v3, v4), (v4, v5) respectively are,
$e1 = 0.125, e2 = 0.375, e3 = 0.625, e4 = 0.875$

For Analyzability, by putting v = 0.372 in (1) - (4), we can get the membership degree as below,
$B1(v) = 0.024, B2(v) = 1 , B3(v) = 0 , B4(v) = 0$

Similarly, the fuzzy evaluation matrix of the 2nd layer sub-criteria is obtained as,

$$F(2) = \begin{bmatrix} 0.024 & 0.008 & 0 & 0 \\ 1 & 1 & 0.012 & 1 \\ 0 & 0 & 1 & 0.576 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus the fuzzy evaluation matrix of the 1st layer criterion is,

$$S_Y = F(2) * w(2) = \begin{bmatrix} 0.003 & 0.382 & 0.677 & 0 \end{bmatrix}'$$

In the same way, software Z can be evaluated as,

$$S_Z = \begin{bmatrix} 0.118 & 0.999 & 0.094 & 0 \end{bmatrix}'$$

According to the maximum membership degree principle [51], we can infer that the result of software product Y is "good" and the result of software product Z is "average". So it can be concluded that Y's maintainability is better than Z's.

# 9 CONCLUSION

This paper developed a methodology that facilitates the qualitative evaluation of object-oriented software systems based on fuzzy approach. Maintainability model as described in ISO/IEC-9126 was taken as the base model. Object-oriented metrics that aid in identifying certain design 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, and thus suggests that it can be a useful and practical addition to the framework of software maintainability assessment. The future work would deal with the determination of the exact value of maintainability and realization of the full potential and possible limitation of fuzzy layered approach.

# REFERENCES

[1] S.R. Ragab, H.H. Ammar, "Object-Oriented design metrics and tools: a survey", *Proc. Of Informatics and Systems (INFOS)*, pp. 1 – 7, Mar. 2010.

[2] R. Glass, *Facts and Fallacies of Software Engineering*, Addison-Wesley Professional, 2002.

[3] R.S. Pressman, *Software Engineering - A Practitioner's Approach*, 7th ed., McGraw Hill, 2005.

[4] A. De Lucia, E. Pompella, S. Stefanucci, "Assessing effort estimation models for corrective maintenance through empirical studies," *Information and Software Technology*, vol. 47, no. 1, pp. 3-15, 2005.

[5] P.S. Roger, *Software Engineering, A Practitioner's Approach*, McGraw Hill International Edition, 2005.

[6] Chen Jing, Liu Xuyan, "Software Maintainability Metrics Based on the Index System and Fuzzy Method", *The 1st International Conference on Information Science and Engineering (ICISE2009)*, pg. 5117-5120, Dec. 2009.

[7] T.L. Saaty, "How to make a decision: The Analytic Hierarchy Process", *European Journal of Operational Research*, vol. 48, pp. 9-26, 1990.

[8] D. Kafura, R. Reddy, "The use of Software Complexity Metrics in Software Maintenance", *IEEE Trans. Software Engineering*, vol. SE-13, no. 3, pp. 335-343, 1987.

[9] S.W.A. Rizvi, R.A. Khan, "Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)", *Journal of Computing*, vol. 2, issue 4, Apr. 2010.

[10] S.R. Chidamber, C.F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, Jun. 1994.

[11] Amandeep Kaur, Satwinder Singh, Dr. K. S. Kahlon, Dr. Parvinder S. Sandhu, "Empirical Analysis of CK & MOOD Metric Suit", *International Journal of Innovation, Management and Technology*, vol. 1, no. 5, Dec 2010.

[12] K.K. Aggarwal, Yogesh Singh, Jitender Kumar Chhabra, "An Integrated Measure of Software Maintainability", *Reliability and Maintainability Symposium*, pp. 235 – 241, 2002.

[13] C.A. Slavila, C. Decreuse, M. Ferney, "Fuzzy Approach for Maintainability Evaluation in the Design Process", *Concurrent Engineering*, vol. 13 no. 4, pp. 291-300, Dec 2005.

[14] L. Zhong, S. Youchao, "Research on Maintainability Evaluation Model Based on Fuzzy Theory", *Chinese Journal of Aeronautics*, vol. 20 (5), pp. 402-407, 2007.

[15] H. Mittal, P. Bhatia, "Software maintainability assessment based on fuzzy logic technique", *ACM SIGSOFT SEN*, vol. 34, issue 3, May 2009.

[16] A. Sharma, P. S. Grover, R. Kumar, "Predicting Maintainability of Component-Based Systems by Using Fuzzy Logic", *Communications in Computer and Information Science*, vol. 40, part 11, pp. 581-591, 2009.

[17] Lizhi Cai, Zhenyu Liu, Juan Zhang, Weiqin Tong, Genxing Yang, "Evaluating Software Maintainability Using Fuzzy Entropy Theory", *IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS)*, pp.737-742, Aug 2010.

[18] Jing Rong Li, Qing Hui Wang, Hong Zhou Shen, "Fuzzy Evaluation of Maintainability with Tribological Factors at Design Stage", *Applied Mechanics and Materials*, pp. 1027-1033, Aug 2010.

[19] Qingbo Hao, Zhaojun Yang, Fei Chen, Binbin Xu, Xiaobing Li, Chuanhai Chen, , "A fuzzy maintainability allocation method for NC machine tools based on interval analysis," *9th International Conference on Reliability, Maintainability and Safety (ICRMS)*, pp.889-896, Jun 2011.

[20] Haiquan Yu, Gaoliang Peng, Wenjian Liu, "A practical method for measuring product maintainability in a virtual environment", *Assembly Automation*, vol. 31 issue 1, pp. 53 – 61, 2011.

[21] M. Zarrin, M.A. Seyyedi, M. Mohsenzaeh, "Designing a Comprehensive Model for Evaluating SOA-based Services Maintainability", *(IJCSIS) International Journal of Computer Science and Information Security*, vol. 9, no. 10, Oct 2011.

[22] K. Jeet and R. Dhir, "Bayesian and Fuzzy Approach to Assess and Predict the Maintainability of Software: A Comparative Study," *ISRN Software Engineering*, vol. 2012, Article ID 202980, 10 pages, 2012.

[23] IEEE Standard Glossary of Software Engineering Terminology, Report IEEE Std. 610.12-1990, IEEE, 1990.

[24] ISO/IEC 9126-1: Software engineering - product quality - part 1: Quality model, ISO Standard, Geneva, Switzerland, 2001.

[25] T.L. Saaty, *The Analytic Hierarchy Process*, New York: McGraw Hill. International, 1980.

[26] R.N. Wabalickis, "Justification of FMS with the Analytic Hierarchy Process", *Journal of Manufacturing Systems*, vol. 17, pp. 175-182, 1988.

[27] P. Putrus, "Accounting for Intangibles in Integrated Manufacturing (nonfinancial justification based on the Analytical Hierarchy Process)", *Information Strategy*, vol. 6, pp. 25-30, 1990.

[28] A. Arbel, Y.E. Orger, "An application of AHP to bank strategic planning: The merger and acquisitions process", *European Journal of Operational Research*, vol. 48 (1) pp. 27–37, 1990.

[29] I.C. Ehie, C.O. Benjamin, Y. Omurtag, L. Clarke, "Prioritizing development goals in low-income developing countries", *Omega*, vol. 18 (2), pp. 185–194, 1990.

[30] T.O. Boucher, E.L. McStravic, "Multi-attribute Evaluation Within a Present Value Framework and its Relation to the Analytic Hierarchy Process", *The Engineering Economist*, vol. 37, pp. 55-71, 1991.

[31] K.E. Cambron, G.W. Evans, "Layout Design Using the Analytic Hierarchy Process", *Computers & IE*, vol. 20, pp. 221-229, 1991.

[32] K.K. Aggarwal, "Software Reliability Apportionment using the Analytic Hierarchy Process", *ACM SIGSOFT SEN*, vol. 20(5), pp. 56, 1995.

[33] K. Park, C.H. Lim, "A structured methodology for comparative evaluation of user interface designs using usability criteria and measures", *International Journal of Industrial Ergonomics*, pp. 379-389, 1999.

[34] W. Ossadnik, O. Lange, "AHP-based evaluation of AHP-Software", *European Journal of Operational Research*, pp. 578-588, 1999.

[35] Hunjak, Tihomir, D. Jakovèeviæ, "AHP based model for bank performance evaluation and rating", *Proceeding of ISAHP 2001*, Berne, Switzerland, 2001.

[36] Michnik, Jerzy, Lo, Mei-Chen, "Enhancement of financial risk management with the aid of analytic hierarchy process", *Proceedings of ISAHP 2005*, Hawaii, 2005.

[37] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, N.Tsirakis, "A Data Mining Methodology for Evaluating Maintainability according to ISO/IEC-9126 Software Engineering-Product Quality Standard", *European Conference on Software Maintenance and Reengineering: Workshop on System Quality and Maintainability*, 2007.

[38] R. Zhou, "How to Quantify User Experience: Fuzzy Comprehensive Evaluation Model Based on Summative Usability Testing", *UI-HCII'07 Proceedings of the 2nd international conference on Usability and internationalization*, pp. 564-573, 2007.

[39] S.S. Kusumaningtuti, P. Abdullah, D. Yumanita, "Legal role on banking crisis resolution in Indonesia: An Analytic Network Process approach", *Proceedings of ISAHP 2007*, Chile, 2007.

[40] Fan, Chiang Ku, Cheng, Shu Wen, "Using Analytic Hierarchy Process Method and Technique for Order Preference by Similarity to Ideal Solution to Evaluate Curriculum in Department of Risk Management and Insurance", *Journal of Social Sciences*, vol. 19 (1), pp. 1-8, 2009.

[41] C. Sanga, I.M. Venter, "Is a Multi-criteria Evaluation Tool Reserved for Experts?", *The Electronic Journal Information Systems Evaluation*, vol. 12 issue. 2, pp165 - 176, 2009.

[42] Y. Kanellopoulos, P. Antonellis, D. Antoniou, C. Makris, E. Theodoridis, C. Tjortjis, N. Tsirakis, "Code Quality Evaluation Methodology Using The ISO/IEC 9126 Standard", *International Journal of Software Engineering & Applications*,vol.1.3, pp. 17-36, Jul 2010.

[43] ZHANG Jun-ping, ZHU Xiaodong, BU Guo-feng, LIANG Xin, YU Na, "Applying Excel in the Weight Calculation of Software Maintainability Evaluation Based on AHP", *International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, vol. 10, Aug 2010.

[44] E. Rasoulinezhad, "Evaluation of countries solution against the U.S economics crisis through a multiple attribute decision model (MADM)", *Journal of Economics and International Finance,* vol. 4 (1), pp. 9–17, Jan 2012.

[45] S. Zokaee, S.B. Ebrahimi, M. Ghazizadeh, "Electronic Payment Systems Evaluation: A Case Study in Iran", *Information Management and Business Review*, vol. 4, no. 3, pp. 120-127, Mar 2012.

[46] S.K. Dubey, A. Rana, "A Comprehensive Assessment of Object-Oriented Software Systems Using Metrics Approach", *International Journal on Computer Science and Engineering*, vol. 02, no. 08, pp. 2726-2730, 2010.

[47] T. Türk, "The Effect Of Software Design Patterns On Object-Oriented Software Quality And Maintainability", M. Tech Thesis, Dept. EE. Eng., Univ. Middle East Technical University, Sep 2009.

[48] S.K. Dubey, A. Rana, "Assessment of Maintainability Metrics for Object-Oriented Software System", *ACM SIGSOFT SEN*, vol. 36, No. 5, Sept 2011.

[49] L. Rosenberg, L. Hyatt, "Software Quality Metrics for Object-Oriented System Environments", *NASA Technical Report SATC,* no. 1, pp 11-58, 2001.

[50] YANG Lunbiao, GAO Yingyi, "Principle and Application of Fuzzy Mathematics", 3rd ed. *South China University of Technology Press.* Guangzhou, China, 2002.

[51] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer-Verlag, Berlin, 1996.