A New Approach of Software Quality Metrics: Fuzzy Extension Algorithm and Its Application

Jianli Dong
School of Computer Engineering, Huaihai Institute of Technology, Lianyungang, 222005, China.
dongjl1019@sina.com

Ningguo Shi
Lanzhou Resources and Environment Vocational and Technical College, Lanzhou, 730021, P.R.China.
sng@lzre.gs.edu.cn

Wenbin Hu
School of Computer Engineering, Huaihai Institute of Technology, Lianyungang, 222005, P.R.China.
hwb1008@163.com

Abstract**—According to the needs and characteristics of software quality comprehensive assessment, the author systematically analyzes the drawback of the loss of a large amount of effective information caused by the calculation in the fuzzy synthesis evaluation algorithm as well as the defects of distortion among the intermediate data transformation in the processing of multi-level extensive evaluation algorithm. For that reason, this paper combines the two original algorithms and improves their shortcomings. Finally, an innovative software quality fuzzy extension synthesis evaluation method is created. Practice shows that adopting this new method to implement software quality evaluation not only has the advantages of accurate, reliable and practical but also the characteristics of easy to achieve computable and standardized.**

Index Terms**—fuzzy synthesis evaluation, software quality, matter-element model, extension evaluation algorithm, evaluation index system**

I. INTRODUCTION

Software quality is the basic guarantee and the lifeline of the development of the software industry. The fundamental goal of software engineering is to produce high-quality software products in the premise of controlling funds and pace. However, after decades of software engineering research, development and application, software quality assurance has not fundamentally resolved. Even now, accidents caused by software quality occurred frequently, especially such heavy casualties and damage in the military, economic, financial and national security applications, the situation of software quality remains worrying all the same. Into the twenty-first century, the development of human society has entered the network and information age. The growing scale and high risk of investment, the complexity and difficulty of production has been the basic characteristics of modern software production and difficult to ensure software quality is the bottleneck that troubled the development of modern software industry. Decades of practice proved that the problems of software quality assurance would not be solved completely overnight due to the special nature of software products, it requires developing constantly new techniques and methods to improve software quality gradually through long-term software engineering research and practice. Thus, software quality assurance is still the difficult and hot issue of the software engineering currently [1, 2].

II. THE STATUS OF SOFTWARE QUALITY METRICS

A. *Some Problems of Existing Metrics Methods*

In the process of the research and development of software engineering, a certain technology, methods and standards have also been gradually developed in software quality measurement and evaluation. Among them the most influential one is the software quality evaluation model put forward by Walters, MaCall, Boehm and others in 1978, as well as a series of software quality evaluation criteria formulated and promulgated by International Organization for Standardization in 1993, including ISO / IEC 14598, ISO / IEC 9126 [3].

The standardized software quality model puts forward various factors and the structure reflecting software products' quality, but how to determine a scientific and accurate method to achieve a quantitative measurement of software quality and the impartiality of evaluation is the key. Objectively speaking, up to now, due to the logical property, abstraction, complexity and large-scale of software products, there is still no a general algorithm of quantitative metrics and evaluation methods of software quality which is recognized and accepted by software engineering field. The current quantitative metrics methods of software quality, such as weighted average method, hierarchy analytic process described in Ref.[4, 5], fuzzy synthesis evaluation method described in Ref.[6, 7]. Furthermore, Chinese expert Wen CAI and his colleagues proposed the matter-element extension evaluation method which is discussed in Ref.[8, 9], etc., only have their own advantages, but also have their own defects which is described in Ref.[10, 11, 12] in detail. So, more accurate, reliable and practical metrics methods have been researched for software quality [13].

The paper will extract the basic calculation structures and novel ideas of the original fuzzy and matter-element extension evaluation algorithm, and improve their respective defects, and finally create a more reliable and

---

This project is supported by Fund of Jiangsu University Natural Science Basic Research Project, Grant No. 08KJD520013.

Corresponding author:Jianli DONG, E-mail: dongjl1019@sian.com

doi:10.4304/jsw.6.10.1881-1889

practical new software quality fuzzy extension synthesis evaluation method. Based on the application results, it can be seen that the new approach can achieve better evaluation for software quality.

B. Software Quality Evaluation Index System.

In order to achieve fuzzy extension synthesis evaluation method for software quality, we need to establish a kind of software quality evaluation index system which must be adapt to extension matter-element model and fuzzy synthesis evaluation as well as the software quality standard measurement model.

Here, following the product-centric quality view, the software quality measurement model and the quality characteristics system put forward by McCall et al. are adopted in this paper [4, 5]. Furthermore, this system is abstracted to three levels to consist of three-class evaluation index system according to composition and affiliation constituting. At the same time, the lower-class index is the refinement and decomposition of the upper-class index, as shown in Table 1. The first-class index (quality characteristics) is the abstract and decomposition of relevant attributes of entire software product quality, which can be divided into product operation, product changes and product modifications and numbered by c_1, c_2 and c_3, respectively.

The three indexes not only reflect the different aspects of software quality attributes independently but also include and summarize all the factors and scope describing software quality. The second-class indexes (sub-characteristics) are the refinement and decomposition of corresponding first-class indexes, which include 11 indexes and are numbered by c_{11}, c_{12}, …, c_{32}, c_{33}, respectively.

The third-class indexes (metric element) is the refinement and decomposition of corresponding second-class indexes, which are atomic indexes that can be directly used for measurement and include 41 indexes that can be numbered by C_{111}, C_{112}, …, C_{332}, C_{333}, respectively. With regard to the calculation of indicators weight at all levels, it should be calculated step by step using the methods such as AHP (Analytic hierarchy process) or rotational expert evaluation method according to the impact of various indexes affecting on software quality or the upper-class index, which should be marked in the brackets after indexes. Thus, the entire software quality evaluation index system is completed.

III. FUZZY AND MATTER-ELEMENT EXTENSION EVALUATION ALGORITHM INTRODUCTION

For creating a new fuzzy extension evaluation method, original matter-element extension valuation algorithm and original fuzzy synthesis algorithm is firstly introduced below. And then their advantages and defects will be analyzed and discussed. Final a new fuzzy extension synthesis valuation algorithm is created [9-11].

TABLE 1. SOFTWARE QUALITY VALUATION INDEX SYSTEM

Software	First Class Index	Second Class Index	Third Class Index
Software Products Quality	Product Operation c_1 (0.4)	correctness c_{11} (0.25)	traceability c_{111}(0.3) completeness c_{112}(0.35) consistency c_{113}(0.35)
		integrity c_{12} (0.2)	access control c_{121}(0.5) access audit c_{122}(0.5)
		efficiency c_{13}(0.15)	execution efficiency $_{c131}$(0.6) storage efficiency c_{132} (0.4)
		usability c_{14} (0.2)	operability c_{141}(0.3) training c_{142}(0.2) communicativeness c_{143}(0.2) i/o capacity c_{145}(0.15) i/o speed c_{146}(0.15)
		reliability c_{15} (0.2)	error-tolerance c_{151}(0.35) consistency $c_{15}2$(0.3) accuracy c_{153}(0.2) simplicity c_{154}(0.15)
	product revision c_2 (0.3)	maintainability c_{21} (0.4)	consistency c_{211}(0.25) conciseness c_{212}(0.15) modularity $c_{21}3$(0.25) self-descriptiveness c_{214}(0.2)
		flexibility c_{22} (0.25)	modularity c_{221}(0.3) consistency c_{222}(0.25) expandability c_{223}(0.25) self-descriptiveness c_{224}(0.2)
		Testability c_{23} (0.35)	simplicity c_{231}(0.2) modularity c_{232}(0.3) self-testability c_{233}(0.3) self-descriptiveness c_{234}(0.2)
	Product transition c_3 (0.3)	portability c_{31} (0.4)	modularity c_{311}(0.2) self-descriptiveness c_{312}(0.2) device-independency c_{313}(0.3) soft-independency c_{314}(0.3)
		reusability c_{32} (0.3)	generality c_{321}(0.15) modularity c_{322}(0.3) self-descriptiveness c_{323}(0.15) device-independency c_{324}(0.2) soft-independency $c_{32}5$(0.2)
		interoperability c_{33} (0.3)	modularity c_{331}(0.3) communication commonality c_{332} (0.35) data commonality c_{333}(0.35)

A. Matter-Element Model of Software Quality Metrics

As shown in Table 1, once software quality evaluation index system is established, the matter-element model for quality characteristics (the first-class index) needed by software quality extension evaluation algorithm can be built. In order to discuss conveniently and without loss of generality, maybe the number of quality characteristics can be assumed to be m (such as in Table 1, m=3), which could be denoted by c_1, c_2, …, c_m respectively. According to the requirements of comprehensive evaluation, software quality evaluation grades can be quantitatively divided into n-levels (such as excellent, good, qualified, unqualified) by software evaluation experts or through statistical analysis methods, and the value range of the evaluation index c_1, c_2, …, cm can also be determined when software quality achieves to different levels. Thereby, the matter-element model of software quality evaluation (also known as "classical domain") can be established as follows:

$$R_{0j}=\begin{bmatrix} N_{0j} & c_1 & V_{0j1} \\ & c_2 & V_{0j2} \\ & \cdots & \cdots \\ & c_m & V_{0jm} \end{bmatrix}$$

$$=\begin{bmatrix} N_{0j} & c_1 & <a_{0j1},b_{0j1>} \\ & c_2 & <a_{0j2},b_{0j2>} \\ & \cdots & \cdots \\ & c_m & <a_{0jm},b_{0jm>} \end{bmatrix}$$

Where j=1, 2, ..., n; R_{0j} denotes the j-class matter-element model of software quality, N_{0j} refers to the j-class software quality, $V_{0jk} = <a_{0jk}, b_{0jk}>$ (k=1, 2, …, m) is the value range of c_k which is the k-th evaluation index while the level of software quality is j, Here, the range can be open, closed or half-open-closed.

In addition, the matter-element model constituted by the entire allowed value range of each index of software quality comprehensive evaluation (also can be called "section field ") can be expressed as:

$$R_p=\begin{bmatrix} N_p & c_1 & V_{p1} \\ & c_2 & V_{p2} \\ & \cdots & \cdots \\ & c_m & V_{pm} \end{bmatrix}=\begin{bmatrix} N_p & c_1 & <a_{p1},b_{p1>} \\ & c_2 & <a_{p2},b_{p2>} \\ & \cdots & \cdots \\ & c_m & <a_{pm},b_{pm>} \end{bmatrix}$$

Where Rp means the section field of matter-element model of software quality comprehensive evaluation, N_p refers to all the grades of software quality evaluation results. $V_{pk} = <a_{pk}, b_{pk}>$ indicates the value range (Value interval) of the index c_k in N_p, $V_{0jk} \subset V_{pk}$, j=1, 2, …, n; k=1, 2, …, m.

For the software to be evaluated, the actual measurement data or analysis results of all indicators can be expressed by the following matter-element model:

$$R=\begin{bmatrix} N & c_1 & v_1 \\ & c_2 & v_2 \\ & \cdots & \cdots \\ & c_m & v_m \end{bmatrix}$$, where k=1, 2, …, m; N refers to the quality of software to be evaluated, v_k indicates the measurement value of the k-th index of software to be evaluated.

B. Matter-Element Extension Evaluation Algorithm

After the establishment of the above matter-element model of software quality comprehensive evaluation, it is necessary to assess the quality of software to be evaluated according to evaluation levels. Therefore, the "correlation" between matter-element model to be evaluated and the classical domain of matter-element model needs to be calculated. In practice, the calculation of "correlation" should be selected on the basis of the characteristics of each index. Here, the elementary correlation function of extenics is used. Suppose that:

$$\rho(v_k,V_{0jk})=\left|v_k-\frac{a_{0jk}+b_{0jk}}{2}\right|-\frac{1}{2}(b_{0jk}-a_{0jk}) \quad ,$$

where k=1, 2, …, m; j=1, 2, …, n.

$$\rho(v_k,V_{pk})=\left|v_k-\frac{a_{pk}+b_{pk}}{2}\right|-\frac{1}{2}(b_{pk}-a_{pk})$$, where k=1, 2, …, m.

They indicate the "correlation" between point v_k and interval V_{0jk}, V_{pk} respectively. For example, $\rho(v_k,V_{pk})\geq 0$, it shows that v_k is not in the range of V_{pk}, while $\rho(v_k,V_{pk})\leq 0$ shows that v_k is within the range of V_{pk}. And various negative values express the different location of v_k in the range of V_{pk}. The suppose that:

$D(v_k,V_{pk},V_{0jk})=\rho(v_k,V_{pk})-\rho(v_k,V_{0jk})$ indicates the location of point v_k between interval V_{0jk} and interval V_{pk}.

$$K_j(v_k)=\frac{\rho(v_k,V_{0jk})}{D(v_k,V_{pk},V_{0jk})}$$, where k=1, 2, …, m; j=1, 2, …, n. It means the correlation degree of the k-th index c_k of matter-element to be valuated and the j-th level evaluation results, $-\infty<K_j(v_k)<+\infty$.

$K_j(v_k)\geq 0$ means that v_k belongs toV_{0jk} and the larger $K_j(v_k)$ is, the more properties of V_{0jk} v_k has; $K_j(v_k)\leq 0$ means that v_k does not belong to V_{0jk} and the smaller $K_j(v_k)$ is, the farther v_k is from the interval V_{0jk}.

Thereby, the correlation matrix of the evaluation indexes of software to be evaluated and various evaluation levels can be calculated: $K=[K_j(v_k)_{m\times n}]$. According to the correlation matrix $K=[K_j(v_k)_{m\times n}]$, it can be calculated as follows:

$$K^*(v_k)=K_{i_0}(v_k)=\max_{1\leq j\leq n}K_j(v_k)$$, k=1, 2, …, m.

Then $K_{i_0}(v_k)$ means the k-th evaluation index of software to be valuated is in the i_0-th level assessment, then the comprehensive evaluation of software quality can be decided by $K_{i_0}(v_k)$. And the approach is described as follows:

Suppose that $\alpha_i(\sum_{i=1}^{m}\alpha_i=1)$ is the weight coefficient of software quality evaluation index, then the correlation degree of software to be evaluated and the j-level evaluation results are:

$$K_j(R)=\sum_{i=1}^{m}\alpha_i K_j(v_{i)})$$, Where j=1, 2, ..., n, then calculating $K_{j_0}(R)=\max_{1\leq j\leq n}K_j(R)$. Thus, we can obtain that the comprehensive evaluation grade of the quality of the software be assessed is at the level of j_0 [9-11].

C. Fuzzy Synthesis Evaluation Algorithm

In order to facilitate the analysis, single-level and multi-level fuzzy synthesis evaluation algorithms are listed below firstly [8] (Note: In the following discussion of fuzzy evaluation algorithm, the variable x has the same meaning as the quality evaluation index c of extension evaluation method. They only retain their own names).

1) Single-Level Synthesis Evaluation Algorithm

a) To Determine the Factor Set of Evaluation Object: $X=\{x_1, x_2, \ldots, x_n\}$;

b) Given the Evaluation Set (Comments Set): $Y = \{y_1, y_2, \ldots, y_m\}$;

c) To Build a Fuzzy Mapping From X to Y:

$\underset{\sim}{f}: X \rightarrow F(Y)$, $x_1 \rightarrow r_{i1}/y_1 + r_{i2}/y_2 + \ldots + r_{im}/y_m$,

where: $0 \le r_{ij} \le 1$, i = 1, 2, …, n; j=1, 2, …, m.

d) To Write the Fuzzy Evaluation Matrix:

$\underset{\sim}{R} = (r_{ij})_{n\times m}$.

e) To Determine the Weight Distribution of Each Factor:

$\underset{\sim}{A} = (a_1, a_2, \ldots, a_n)$, where a ≥ 0, $\sum_{i=1}^{n} a_i = 1$.

f) To Implement Comprehensive Evaluation with Fuzzy Matrix:

$B = \underset{\sim}{A} \cdot \underset{\sim}{R} = (b_1, b_2, \ldots, b_m)$, where:

$$b_j = \bigvee_{i=1}^{n} (a_i \wedge r_{ij}),\ j = 1, 2, \ldots, m. \quad (1)$$

And to seek the result of max{$b_1, b_2, \ldots, b_m$}$=b_{j_0}$, j_0□{1, 2, …, m}.

Evaluation conclusion: The review of the object being evaluated is b_{j_0}.

2) Multi-level Synthesis Evaluation Algorithml

Multi-level synthesis evaluation algorithm is mainly used for the comprehensive evaluation of complex issues (system). The principle is to decompose and refine the evaluation factors from the higher to the lower level on the basis of relevant attributes to form factors' multi-level affiliation, and then judging from the lowest to the high level using single-level synthesis evaluation algorithm and stopping until the highest level (evaluation objects) achieved the conclusion. The algorithm model is as follows:

a) To Divide Factor Set:

The factor set $X=\{x_1, x_2, \ldots, x_n\}$ can be divided into s sub-sets according to their element attributes, denoted as: $X_1, X_2, \ldots, X_s$, where $X_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{q(i)}}\}$, $X_i \cap X_j = \phi$ (i ≠ j; i, j = 1, 2, …, s), $\bigcup_{i=1}^{s} X_i = X$, $\sum_{i=1}^{s} i_{q(i)} = n$.

b) To Get the Evaluation Results of Each X_i According to Signal-Level Synthesis Evaluation Algorithm:

Let evaluation set be $Y = \{y_1, y_2, \ldots, y_m\}$, the weight distribution of each factor centralized by X_i is $\underset{\sim}{A}_i = (a_{i_1}, a_{i_2}, \ldots, a_{i_{q(i)}})$, where $\sum_{k=1}^{q(i)} a_{i_k} = 1$, ($a_{i_k} \ge 0$), and the single-level fuzzy evaluation matrix of X_i is $\underset{\sim}{R}_i$. Thus, the single-level synthesis assessment result of X_i can be drawn: $B_i = \underset{\sim}{A}_i \cdot \underset{\sim}{R}_i = (b_{i1}, b_{i2}, \ldots, b_{im})$ (i =1, 2, …, s), Where:

$$b_{ij} = \bigvee_{i=1}^{q(i)} (a_{i_k} \wedge r_{i_k j}),\ (i=1, 2, \ldots, s; j=1, 2, \ldots, m) \quad (2)$$

c) Single-Level Synthesis Evaluation:

using each X_i as the element to form the set $\{X_1, X_2, \ldots, X_s\}$ once again and adopting B_i (i =1, 2, …, s) to be the row vector of fuzzy evaluation matrix $\underset{\sim}{R}^*$ of the advanced signal-level synthesis evaluation of these factors. That is $\underset{\sim}{R}^* = (B_1, B_2, \ldots, B_s)^T = (b_{ij})_{s\times m}$ is the fuzzy evaluation matrix of the factor set $\{X_1, X_2, \ldots, X_s\}$. Taking the weight distribution of the factor set $\{X_1, X_2, \ldots, X_s\}$ as $\underset{\sim}{A}^* = (a_1, a_2, \ldots, a_s)$, then, the synthesis evaluation results can be obtained: $B^* = \underset{\sim}{A}^* \cdot \underset{\sim}{R}^* = (b_1, b_2, \ldots, b_m)$, where:

$$b_j = \bigvee_{i=1}^{s} (a_i \wedge r_{ij}),\ (j =1, 2, \ldots, m) \quad (3)$$

d) To Calculate Final Fuzzy Synthesis Evaluation:

Repeating step b) and c) until obtain the fuzzy synthesis evaluation value of first-level index.

IV. CREATING FUZZY EXTENSION EVALUATION ALGORITHM

The two algorithms described above, the fuzzy and extension evaluation algorithms, separately used to evaluate software quality will produce some adverse effects. In [10, 12], all defects are described. For this reason, we extract the advantages and create a new fuzzy extension synthesis evaluation algorithm, which is now described below.

1) To Determine the Evaluation Index System and grade Set of the Objec Evaluated:

To determine the evaluation index system (as shown in Table 1) of software quality to be evaluated and evaluation set Y based on the standard model of software quality measurement.

2) To Determine the Weights of Evaluation Indexes at All Levels:

The weight of each evaluation index is achieved through AHP, Delphi (Delphi Method) and other mathematical statistical methods. It shouldn't be freely given subjectively. As shown in Table 1, the weight is offered through the three cycles of statistics of Delphi method.

3) To Establish the First-Class Indexes of Matter-Element Model:

Under the guidance of evaluation experts in the field of software, the classical domain and section field matter-element model of the first-class index (i.e. the highest level index) is established. The first-class matter-element model to be evaluated would be calculated and directly obtained by fuzzy synthesis evaluation method.

4) To Establish the Matter-Element Model of the Third-Class Indexes:

To build the classical domain matter-element model of the third-class indexes (metric element).

5) To Change Fuzzy Matrix Operations to the General Matrix Operations:

In order to effectively implement integrated measurement in the comprehensive evaluation algorithm and prevent the loss of a large amount of valid information, fuzzy matrix synthesis operations defined by the formula $b_j = \bigvee_{i=1}^{s} (a_i \wedge r_{ij})$ in the comprehensive evaluation algorithm model can be changed to the ordinary matrix operations defined by the formula $b_j = \sum_{i=1}^{n} (a_i \cdot r_{ij})$ (j=1, 2, …, m). Essentially, it is to identify the calculation of "selecting large and small value" defined by the formula (1), (2) and (3) introduced in signal-level and multi-level comprehensive evaluation algorithm model by the "matrix multiplication" operation

6) To Determine the Fuzzy Membership Function of the Third-Class Indexes:

After determining the classical domain matter-element model of the third class indexes (metric element), the measuring value of each of third-class indexes can be obtained by metric element measurement tools and formulas provided by RADC (Rome Air Development Center) [15]. These measurement values of metric element are original from the most objective, accurate and basic atomic attribute values of evaluation software, but they can not be used for the fuzzy membership values of metric element to evaluation grades. Therefore, according to the distribution of metric element measurement values of software quality and the evaluation experience, the membership function of metric element for different levels of evaluation can be regarded as normal distribution function:

$$\mu(x) = e^{-(\frac{x-m}{c})^2} \qquad (4)$$

Where: m and c are constants, the value range of variable x is restricted to the classical domain interval of matter-element model of third class indexes, μ(x) is used to determine the fuzzy membership value of the third-class indexes measurement values at classical domain evaluation grades. The definition of constant m and c:μ(x)=1 and the value of membership degree is maximum, therefore, m could be see as the middle-point value (average value) of the interval of classical domain, that is, $m=(x_l+ x_r) / 2$. x_l, x_r are the left and the right end of the third-class indexes classical domain interval. Furthermore, deduced by the formula μ (x), when u (x) is at the critical point of classical domain value range of matter-element model of two adjacent evaluation levels, two membership degree values for two adjacent reviews should be the same, approximately equal to 0.5. That is, when $e^{-(\frac{x_r - x_l}{2c})^2} = 0.5$, we can obtain the value of c in different classical domain range.

So far, the values of constants m and c of fuzzy membership function μ (x) in different interval of third class indexes classical domains have been determined. With RADC method, the specific measurement value of each third-class indexes can be obtained and then fuzzy membership value of the third-class indexes for different evaluation levels can be get through the formula (4). Thus, fuzzy membership matrix of the third-class indexes is available.

7) To Implement Fuzzy Evaluation Calculation From the Lower to the Higher Level:

After the fuzzy membership matrix of the third-class indexes is determined, on the basis of multi-level fuzzy synthesis evaluation the fuzzy comprehensive assessment could be carry out layer by layer from metric element (the third-class indexes) to quality characteristics (the first-class indexes) until the quality evaluated value of the first-class indexes (quality characteristics) in the software quality evaluation system, thus, the matter-element model to be evaluated of software quality can be available. In other words, the specific measurement values of quality characteristics are calculated level by level from the bottom to the top through fuzzy synthesis evaluation approach.

8) To Implement the Extension Algorithm Evaluation:

When the matter-element model to be evaluated of the first-class indexes is established, one can implement software quality comprehensive evaluation using matter-element model extension assessment method from the first-class indexes of the evaluation system for a start.

Up to now, fuzzy extension synthesis evaluation method of software quality has been accomplished.

V. THE APPLICATION OF THE FUZZY EXTENSION SYNTHESIS EVALUATION FOR SOFTWARE QUALITY

The application example of the fuzzy extension synthesis evaluation method for software quality is addressed below.

1) To Determine the Evaluation Index System and Grade Set of Software Quality to Be Evaluated:

The evaluation index system of software quality being evaluated is shown in Table 1. The weights of each index have also been given. Evaluating grade set Y= (excellent, good, fair, poor, bad).

2) To Establish Matter-Element Models of the First-Class and the Third-Class Indexes:

Due to space limitations and understanding easily, the authors demonstrates the calculation process of fuzzy extension algorithm only by establishing matter-element models depending on parts of indexes. Suppose that the classical domain and section field of matter-element model of the first-class indexes are as follows (Note: in order to meet the practices of matter-element model and fuzzy evaluation, we take c as the name of index in matter-element model and take x as the name of index in the fuzzy evaluation. In fact, they are the same. As shown in Table 1, it can be considered that $c_1=x_1$, $c_2=x_2$, …, $c_{333}=x_{333}$):

$$R_{01}=\begin{bmatrix} N_{01} & c_1 & <0.82,1> \\ & c_2 & <0.83,1> \\ & c_3 & <0.82,1> \end{bmatrix},$$

$$R_{02}=\begin{bmatrix} N_{02} & c_1 & <0.62,0.82> \\ & c_2 & <0.70,0.83> \\ & c_3 & <0.60,0.82> \end{bmatrix},$$

$$R_{03}=\begin{bmatrix} N_{03} & c_1 & <0.45,0.62> \\ & c_2 & <0.50,0.70> \\ & c_3 & <0.46,0.60> \end{bmatrix},$$

$$R_{04}=\begin{bmatrix} N_{04} & c_1 & <0.35,0.45> \\ & c_2 & <0.36,0.50> \\ & c_3 & <0.35,0.46> \end{bmatrix},$$

$$R_{05}=\begin{bmatrix} N_{05} & c_1 & <0,0.35> \\ & c_2 & <0,0.36> \\ & c_3 & <0,0.35> \end{bmatrix}.$$

$$R_p=\begin{bmatrix} N_p & c_1 & <0,1> \\ & c_2 & <0,1> \\ & c_3 & <0,1> \end{bmatrix}.$$

The classical domain value interval of matter-element model of the third-class indexes c_{111}, c_{112} and c_{113} are shown in Table 2.

TABLE 2. THIRD CLASS INDEXES MATTER ELEMENT MODEL

third class indexes	excellent	good	fair	poor	very poor
c_{111}	(0.85, 1)	(0.75,0.85)	(0.55,0.75)	(0.40,0.55)	(0,0.40)
c_{112}	(0.86, 1)	(0.74,0.86)	(0.57,0.74)	(0.35,0.57)	(0,0.35)
c_{113}	(0.84, 1)	(0.75,0.84)	(0.60,0.75)	(0.40,0.60)	(0,0.40)

TABLE 3. THIRD CLASS INDEX NORMAL DISTRIBUTION PARAMETER

third class indexes	excellent	good	fair	poor	very poor
	m, c	m, c	m, c	m, c	m, c
c_{111}	0.93,0.09	0.80,0.06	0.65,0.12	0.48,0.09	0.20,0.20
c_{112}	0.93,0.08	0.80,0.07	0.66,0.10	0.46,0.13	0.18,0.21
c_{113}	0.92,0.10	0.80,0.05	0.68,0.09	0.50,0.12	0.20,0.24

According to the calculation method of constants m and c in the formula (4), the constant values of every classical domain interval corresponded to m and c can be obtained and are shown in Table 3.

3) To Determine the Fuzzy Membership Matrix of the Third-Class Indexes:

According to multi-level fuzzy comprehensive evaluation process, the fuzzy evaluation matrix corresponding to the third-class indexes (metric element) in software quality evaluation index system (Table 1) must be determined firstly. The elements of fuzzy matrix are made of each membership degree of elements in conclusion set Y. Here, evaluation factors measurement algorithm studied by U.S. Air Force RADC (Rome Air Development Center) is adopted. Then using software measurement tools to measure actually, actual measured values of the third-class indexes can be available. Due to space limitation, here only " c_{111} = the traceable demand number / the total demand number, c_{112}= the sum of ratings meeting the terms of completeness / the total number of terms, c_{113}= 1 - the module number violating rules / the module number" are given as the calculation example. The measurement of the remaining third-class indexes will not be repeated here, please refer to the definition of RADC data.

Through the metric elements measuring formula of RADC, we can obtain that $c_{111}=0.73$, $c_{112}=0.89$, $c_{113}=0.65$. Then, according to Table 2, Table 3 and formula (4), membership values for each evaluation grade can be obtained, which are shown in Table 4.

TABLE 4. THIRD CLASS INDEXES MEMBERSHIP DEGREE

third class indexes	excellent	good	fair	poor	very poor
r_{ijk}	u(x)	u(x)	u(x)	u(x)	u(x)
$c_{111}=0.73$	0.01	0.26	0.64	0.00	0.00
$c_{112}=0.89$	0.80	0.21	0.01	0.00	0.00
$c_{113}=0.65$	0.00	0.00	0.93	0.21	0.03

4) To Evaluate the First-Class Indexes From the Lower to the Higher Level:

Fuzzy membership matrix of the third-class indexes are c_{111}, c_{112} and c_{113}, which can be directly obtained from Table 4. It is as follows:

$$\underset{\sim}{R}_{11}=\begin{bmatrix}0.01 & 0.26 & 0.64 & 0.00 & 0.00\\ 0.80 & 0.21 & 0.01 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.93 & 0.21 & 0.03\end{bmatrix}$$

From the weights of the index system in Table 1, it can be available that $\underset{\sim}{A}_{11}=(0.30,0.35,0.35)$.

Fuzzy evaluation values of the second-class indexes can be obtained through the fuzzy matrix. As regards to c_{11}, the fuzzy evaluation result of c_{11} can be got through the following formula.

$$\underset{\sim}{A}_{11}\bullet\underset{\sim}{R}_{11}=(0.3,0.35,0.35)\cdot$$

$$\begin{bmatrix}0.01 & 0.26 & 0.64 & 0.00 & 0.00\\ 0.80 & 0.21 & 0.01 & 0.00 & 0.00\\ 0.00 & 0.00 & 0.93 & 0.21 & 0.03\end{bmatrix}$$

$$=(0.28,0.15,0.52,0.07,0.01)$$

Here, the evaluation results can be directly used as fuzzy membership value of the second-class index c_{11}. Furthermore, fuzzy membership values of all the second-class indexes can be obtained in this way. Merging the membership values of the second-class index related to the first-class indexes into a fuzzy membership matrix and selecting the weights of the second-class indexes from Table 1, the fuzzy evaluation results of the first-class indexes can be achieved according to fuzzy operator. Take the first-class index c_1 for example, the fuzzy membership matrix resulting in the combination of the second-class index c_{11}, c_{12} , c_{13}, c_{14} and c_{15} can be expressed as follows:

$$\underset{\sim}{R}_{1}=\begin{bmatrix}0.28 & 0.15 & 0.52 & 0.07 & 0.01\\ 0.22 & 0.62 & 0.12 & 0.05 & 0.01\\ 0.12 & 0.63 & 0.18 & 0.06 & 0.02\\ 0.14 & 0.16 & 0.62 & 0.05 & 0.03\\ 0.12 & 0.65 & 0.15 & 0.06 & 0.02\end{bmatrix}.$$

It can be resulted from Table 1 that the weight matrix of c_{11}, c_{12}, c_{13}, c_{14}, c_{15} and c_{16} is $\underset{\sim}{A}_{1}=(0.25,0.20,0.15,0.20,0.20)$. Then, the fuzzy calculation results of c_1 can be obtained by $\underset{\sim}{A}_{1}\bullet\underset{\sim}{R}_{1}=(0.18,0.42,0.34,0.06,0.02)$. Assumed that the fuzzy results of indexes c_2, c_3 calculated by the similar method is (0.15, 0.48, 0.30, 0.05, 0.02) and (0.12, 0.26, 0.44, 0.14, 0.04), respectively. Then, the maximum the calculation results and its corresponding evaluation level are taken as the fuzzy calculated values of the first-class indexes, which is logoed by c_{0jk}. 0j stands for the assessment level, k refers to c_k. Fuzzy calculated value of index c_1 is max(0.18, 0.42, 0.34, 0.06, 0.02) = 0.42=c_{021}.

In order to maintain the consistency among fuzzy calculated value, matter-element model and size of extension data an avoid the data distortion in the transmission and calculation process of fuzzy extension algorithm, the formula, $v_k=a_{0jk}+(b_{0jk}-a_{0jk})\ /\ c_{0jk}$, can be used to obtain measurement values of matter-element model to be evaluated of the first-class indexes. Where, v_k means the measurement value of c_k, (b_{0jk}, a_{0jk}) is the range of the classical domain of matter-element model that c_k corresponded at the grade of the 0j. Then, the measurement value of c_1 is 0.62+(0.82-0.62)*0.42=0.704. Similarly, the value of c_2 and c_3 could be calculated. The matter-element model to be evaluated may be formed as follows:

$$R=\begin{bmatrix}N & c_1 & 0.704\\ & c_2 & 0.762\\ & c_3 & 0.522\end{bmatrix}.$$

5) To Calculate Final Extension Evaluation Results

After determining the classical domain, section field and matter-element model to be evaluated of the first-class indexes by the above method, the matter-element model extension algorithm can be applied to evaluate software quality. The correlation matrix obtained by using correlation function is:

$$K=[K_j(v_k)]_{3\times5}$$

$$=\begin{bmatrix}-0.28 & 0.40 & -0.22 & -0.46 & -0.55\\ -0.22 & 0.35 & -0.21 & -0.52 & -0.63\\ -0.38 & -0.14 & 0.15 & -0.12 & -0.27\end{bmatrix}$$

Thus available: $K^*(v_1)=K_2(v_1)=0.40$, $K^*(v_2)=K_2(v_2)=0.35$, $K^*(v_3)=K_3(v_3)=0.15$. Thus, the evaluation conclusions of each quality characteristics can be acquired, including per-formability, modifiability which is the second-level (i.e. good) and transferability which is the third-level (i.e. fair or middle).

From Table 1, the weights of the first-class indicators are, respectively, A = (0.40, 0.30, 0.30). Then, using A • K = (-0.29, 0.22, -0.11, -0.38, -0.49), obviously, we can know the correlation degree of evaluation software are $K_1(R)=-0.29$, $K_2(R)=0.22$, $K_3(R)=-0.11$, $K_4(R)=-0.38$, $K_5(R)=-0.49$. May be we can select $K_2(R)=\max\limits_{j=1,2,3,4,5}K_j(R)=0.22$. Then, the comprehensive evaluation grade of the software could be measured out quantitatively, which is at level 2, that is, the total rated as good.

6) The Indication of Software Quality Evaluation Results.

The representation and analysis of the final comprehensive evaluation conclusion of the evaluated software quality is an important work in the later period of software quality evaluation. Generally speaking, software quality evaluation results are very important for the demand side and development side because the correct and effective denotation and comprehensive analysis of the evaluation conclusion can help to make a clear understanding of the quality in various aspects, which is essential to strengthen software management, improve the ability of software development, enhance the quality of software development and competitiveness in the market. Currently, software quality quantitative measurement evaluation results has many kinds of expression forms, such as digital representation of percentile, text description ranked in excellent, good, qualified and unqualified. In this paper, the radar figure (Kiviat) which is shown in Fig.1 is used to express it graphically (the solid dots represent the evaluation grades of quality characteristics, the solid five-pointed stars represent comprehensive evaluation grades of software quality), which provide relevant assessment staff an intuitive and effective method to examine comprehensive evaluation result of software quality and the quality level of each quality characteristic.

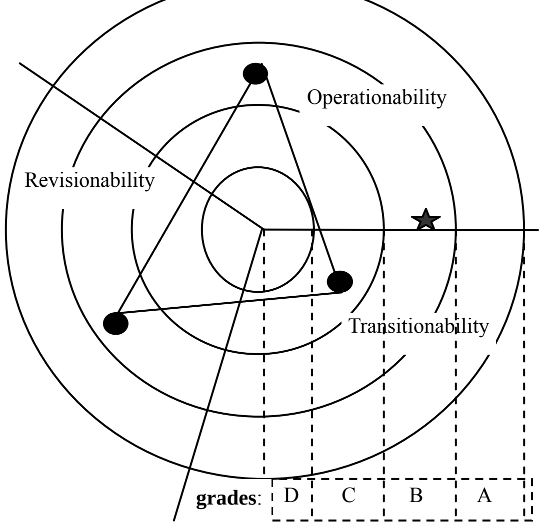

Figure 1. Kiviat Figure of Evaluation Result

VI. CONCLUSION

It should be noted that when simply using matter-element model and extension evaluation method to implement quantitative measurement and evaluation of software quality, the key is to build various matter-element models of extension, namely classical domain, section domain and matter-element model to be evaluated. However, with the increasing levels of quality evaluation index system and numbers of indexes, when creating extension matter-element model and calculating level by level and moving evaluation results, the weight distribution, the identification of correlation function calculation match, the division of evaluation grading as well as the determination of indexes' value range, which are matching to such calculation process, must be taken into consideration overall and quantified. It seems to be particularly complex and cumbersome. But the fuzzy comprehensive evaluation method has the same computing capacity and evaluation effectiveness, in addition, its transmission of calculation level by level and moving is simpler and easier. Therefore, in this aer extension evaluation method and fuzzy evaluation method are merged into a new fuzzy extension evaluation method. It draws the advantages of two algorithms and abandons the shortcomings, after all, it is a better method for software quality comprehensive evaluation [14].

REFERENCE

[1] YANG FuQing, "Thinking on the Development of Software Engineering Technology," Journal of Software (in Chinese), 2005, 16(1), pp.1-7.

[2] BEN Keron, "Outlined Of Software Quality Assurance Technologies Research (in Chinese)", Journal of Naval University of Engineering, 2002, 14(4), pp.1-6.

[3] YIN Ping, "The Analyze of Software Quality Evaluation Based on ISO(in Chinese)," Software Engineering and Standardization, 2005, No.12, pp.37-41.

[4] LIANG Chengcai, TANG Wei, XIAO Liwen, "A Quantitative Assessing Method for Software Quality(in Chinese)," Computer Engineering, 2003, Vol.29(14), pp.95-97.

[5] Stephen H.Kan, "Metrics and Models in Software Quality Engineering (in Chinese)," Boston Addison-Wesley, 2003.

[6] CAI Xiyao, "On Software Quality Metrics (in Chinese)," Computer Engineering & Science, 1996, No.1, pp.1-6.

[7] DONG Jianli, SHI Ningguo, "Research and Improvement of the Fuzzy Synthesis Evaluation Algorithm Based on Software Quality (in Chinese)," Computer Engineering and Science, 2007, Vol.29, No.1, pp66-69.

[8] CAI Wen, "Introduction of Extenics. System Engineering Theory & Practice," 1998, No.1, pp.76-84.

[9] YANG Chunyan, CAI Wen, Extension Engineering(in Chinese). Beijing: Science Press, 2007.

[10] Dong Jianli, "Research and Application of Extension Evaluation Algorithm for Software Quality," IEEE Computer Society, ICIECS 2009, Vol.1, pp.2527-2531.

[11] Dong Jianli, Ningguo Shi, "Multilayer Matter-Element Extension Synthesis Evaluation of Software Quality," IEEE, ICBECS 2010, pp.693-697.

[12] Dong Jianli, Ningguo Shi, "An Improved Fuzzy Synthesis Evaluation Algorithm for Software Quality," IEEE Computer Society, ICIII 2009, Vol.2, pp.565-569.

[13] XU Guanghui, JIN Fenglin, DING Li, "The Method and Practice of Software Engineering," Beijing: China Machine Press, 2004, pp.186-191.

[14] Li Kewen, Gong Lina, Kou Jisong, Predicting software quality by fuzzy neural network based on rough set , Journal of Computational Information Systems, v 6, n 5, p 1439-1448, May 2010.

[15] AD-A153 990, SPECIFICATION OF SOFTWARE QUALITY ATRIBUTES Software Quality Evaluation Guidebook. RADC-TR-85-37, Vol.□-□, Final Technical Report, 1985.

Jianli DONG was born in Shanxi province, China, in 1957. He got his Bachelor of Mathematics Science in Northwest Normal University, Lanzhou, Gansu province, China, in 1988 and got his Master of Software Engineering in Beijing University of Aeronautics and Astronautics, Beijing, China, in 1995. He is now a professor at the School of Computer Engineering in HuaiHai Institute Technology, Lianyungang, China. He has published 40 papers, and completed some research projects, and won 6 times scientific and technological progress awards from the province and military.

Mr. DONG current research interests include software engineering, integrated software engineering environment, software architecture, engineering database system, object-oriented technology.

Ningguo SHI was born in Linxia of Gansu province of China on 11, 1963. He graduated from Xi'an Mining Institute majoying in mine survey in Bechelor Degree in 7, 1985. Now he is a president and a professor in Lanzhou Resources & Environment Voc-Tech College. Pro. Shi is one of first leading talents of Gansu province. Also he is an expert who is entitled to speical grants from the National Council of China, an evaluation expert of Ministry of Education for assessing the talent cultivating work in higher vocational colleges.

Mr. SHI current research interests focuses on constructing and developing an excellent course--Mine Survey, and computer technology and application, software engineering, database system, information technology, etc.