# SOFTWARE QUALITY EVALUATION MODEL

LU JIE
Department of Economy
Hebei University
Baoding, Hebei
China

Abstract

In this paper, we introduce ISO software quality metrical criteria as software evalution index system and multifactorial evaluations technique to establish software quality evaluation model, thus provides a software quality metrical method.

Keywords: Multifactorial evaluation, Evaluation index system, Software quality metrics.

## 1 Introduction

W With development of computer science and appearance of more and more software, software managers and technical personnels in computer now pay a great attentions to the problem of software quality evaluation. In recent years, the evalution of software quality not only is approached in terms of theory but also is widely used in software development, management and acceptance.

Because (1) complexity of software evaluation index system; (2) some evaluation factors cannot be quantitatively described;

(3) each of the factors has its evaluation criteria, therefore it is difficult to synthesize them; (4) the difference in importance of all factors must be expressed in the result of software evaluation; (5) the result of software evaluation needs to be expressed only in gradational form, for example, "A", "B", "C", "D", it is tapically a fuzzy concept, we think it is feasible and effective to make use of fuzzy multfactorial judgement technique to solve problem of software quality evaluation. This paper briefly discribes ISO software quality evaluation index system and uses fuzzy multifactorial judgement technique to establish software quality evaluation model, thus provides a new evaluation method overall software quality and its individual characteristics.

## 2 Software Quality Evaluation Index System

We introduce ISO software quality metrical criteria in 1985 as software evaluation index system.

1) correctness
- traceability
- completeness
- consistency

2) reliability
- accurary
- consistency
- error-tolerance
- simplicity

3) efficiency
- execution efficiency
- storage efficiency

4) security
- access control

access audit

5) usability
- operabitity
- training
- communicativeness

6) maintaine-ability
- consistency
- simplicity
- modularity
- instrumentation
- self-descripliveness
- conciseness

7) flexibility
- modularity
- generality
- expandability
- self-descriptiveness
- software system independence
- machine independence

8) interoper-ability
- communications commonality
- data commonality

3 Mathematics Model

Let first level software quality evaluation index of above system be $a_1, a_2, \cdots, a_8$.

Let evaluation factors( second level ) of software quality evaluation index $a_i$ ( $i = 1, 2, \cdots, 8$ ) be $b_{i1}, b_{i2}, \cdots, b_{ij(i)}$, and call

$$V = \{ b_{ij};\ i = 1, \cdots, 8,\ j = 1, \cdots, j(i) \}$$

evaluation factor set.

We establish mark set according to the requirement. In this paper, mark set is

$$U = \{A, B, C, D\}.$$

Let weight of $a_i$ be $x_i$ ($i = 1, \cdots, 8$) and write

$$X = (x_1, x_2, \cdots, x_8),$$

where $x_i \geq 0$, for the purpose of simplifing calculating process lat $x_i \leq 1$ ($i = 1, \cdots, 8$), $\sum_{i=1}^{8} x_i = 1$.

Similarly, let weight of each evaluation factor $b_{ij}$ of evaluation index $a_i$ ($i = 1, \cdots, 8$) be $y_{ij}$ ($j = 1, \cdots, j(i)$) and write

$$Y_i = (y_{i1}, y_{i2}, \cdots, y_{ij(i)}) \quad i = 1, \cdots, 8,$$

where, $0 \leq y_{ij} \leq 1$ and $\sum_i \sum_j y_{ij} = 1$.

On the basis of above-mentioned, we follow the steps below to compute software quality evaluation.

1) Choose one element of mark set for each evaluation factor $b_{ij}$(($i = 1, \cdots, 8$, $j = 1, \cdots, j(i)$)

Each expert in software quality evaluation groups choose any one element of mark set U for each evaluation factor $b_{ij}$ ($i = 1, \cdots, 8$, $j = 1, \cdots, j(i)$), then calculate subordinate degree of each mark of $b_{ij}$, i.e., the proportion of each mark of $b_{ij}$

$$r_{ij} = (r_{ij}^1, r_{ij}^2, r_{ij}^3, r_{ij}^4) \quad i = 1, \cdots, 8, \ j = 1, \cdots, j(i),$$

where $r_{ij}^1$, $r_{ij}^2$, $r_{ij}^3$, $r_{ij}^4$ are on behalf of the proportions that mark of $b_{ij}$ is A, B, C, D respectively.

2) Determine the mark of each evaluation index $a_i (i=1,\cdots,8)$

$$R_i = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{i\,j(i)} \end{bmatrix} = \begin{bmatrix} r_{ij}^1 & \cdots & r_{ij}^4 \\ & \vdots & \\ r_{i\,j(i)}^1 & \cdots & r_{i\,j(i)}^4 \end{bmatrix} \quad i = 1, \cdots, 8.$$

We make fuzzy matrix opration of $R_i$ and $Y_i$ ( $i = 1,\cdots, 8$),

and obtain evaluation result for $a_i$:

$$N_i = Y_i \circ R_i$$

$$= \left( \overset{j(i)}{\underset{j=1}{\ast}} ( y_{ij} \ast r_{ij}^1 ), \cdots , \overset{j(i)}{\underset{j=1}{\ast}} ( y_{ij} \ast r_{ij}^4 ) \right)$$

$$= ( n_i^1, n_i^2, n_i^3, n_i^4 ) \quad i = 1,\cdots, 8,$$

where $n_i^1$ represents the proportion that mark of $a_i$ is A.

Thereform, we extract

$$n_i^* = \max ( n_i^1, n_i^2, n_i^3, n_i^4 ) \quad i = 1,\cdots, 8,$$

the mark corresponding to the $n_i^*$ is evaluation result for index $a_i$.

3) Determine mark of overall software quality

We eatablish overall software quality evaluation matrix

$$N = \begin{bmatrix} N_1 \\ \vdots \\ N_8 \end{bmatrix} = \begin{bmatrix} n_1^1 & \cdots & n_1^4 \\ & \vdots & \\ n_8^1 & \cdots & n_8^4 \end{bmatrix}$$

then make fuzzy matrix operation of N and X and obtain evaluation result of overall software quality

$$M = X \circ N$$

$$= ( \overset{8}{\underset{i=1}{*}} ( x_i * n_i^1 ), \cdots , \overset{8}{\underset{i=1}{*}} ( x_i * n_i^4 ))$$

$$= ( m_1, m_2, m_3, m_4 )$$

where $m_1$ represents the proportion that mark of overall software quality is A. Therefore, we obtain

$$m^* = \max ( m_1, m_2, m_3, m_4 ),$$

the mark corresponding to the $m^*$ is mark of overall software quality.

4) Compare and appraise the software qualities of different types

When some softwares of different types need comparing and appraising, firstly, we want to determine the weight of each evaluation index and each evaluation factor on type of software. If a factor (or index) is important for the software of certain type, then the factor must have a big weight. If a factor is no importance for a software of certain type, then the weight of this factor can be quite small or zero. Secondly, according to evaluation factor set and mark set we choose marks for every factors and calculate subordinate degree, thus obtain evaluation result of individual characteristics of each software

$$N_i = ( n_i^1, n_i^2, n_i^3, n_i^4 ) \quad i = 1, \cdots, 8,$$

and evaluation result of overall software quality

$$M = ( m_1, m_2, m_3, m_4 ).$$

Lastly, compare $n_i^1 (i = 1, \cdots, 8)$ of every software, the software corresponding to the greatest $n_i^1$ is the best in the individual quality, compare $m_1$ of every softwares, the software corresponding to the greatest $m_1$ is the best in oveall quality.

In this paper, the operational symbol $\overset{+}{*}$ and $\overset{\cdot}{*}$ can represent different calculations, i.e., various mathematics models can be made by model $M(\overset{+}{*}, \overset{\cdot}{*})$, for example, $M(\wedge(\min), \vee(\max))$, $M(\cdot, \vee)$ and $M(\wedge, \oplus)$.

## 4 Conclusion

The model overcomes some one-sidednesses and weaknesses of former software evaluation methods and raises scientific property of software evaluation. Not only can the model be used to grade the software on the basis of its quality (for example, meeting of appraise the software quality), but it also can be used to compare and appraise the qualities of different softwares (for example, decide on awards through discussion for different softwares), it has good practical value.

## References

[1] Chen Yongyi etc., Models of multifactorial evaluations, Fuzzy mathematics, 1(1983), 61-70(China).

[2] Boehm et al., Characteristics of software quality, North Holland, 1978.

[3] Murine G. E. et al., Measuring software product quality, Quality Proress, 5(1984).