International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.5 (2015), pp.111-122
http://dx.doi.org/10.14257/ijmue.2015.10.5.12

Software Project Quality Evaluation Model for Web Application

WanJiang Han[1], WeiJian Li[2], HeYang Jiang[2], TianBo Lu[1] and Xiaoyan Zhang[1]

[1]School of Software Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China
[2]International School, Beijing University of Post and Telecommunication Beijing 100876, China
hanwanjiang@bupt.edu.cn

Abstract

Quality is a major challenge throughout software projects, it is important to analyze and assess quality of a software project. Regular quality measurement is key when it comes to running a successful project. This paper presents a quantitative quality evaluation model of software project for Web application. This model uses Analytical Hierarchy Process (AHP), Fuzzy Mathematical theory and expert method to constrain quality metrics with respect. This model provides the weights of quality characteristics and sub-characteristics using analytic hierarchical process technique, which is on the basis of summering and analysis of historical test data for Web application. The experimental results indicate that this approach is effective for software project quality evaluation and can get good estimates.

Keywords: *Quality Evaluation Model, Analytical Hierarchy Process, Fuzzy Mathematical theory, Web Application, Software project quality*

1. Introduction

Over the last few decades, the problem of project evaluation has attracted significant attention that has led to a variety of methods. These methods seek to develop quantitative measures to assess the projects performance by systematically obtaining and integrating subjective and objective data.

In 1958, Rubey and Herfwick proposed a concept that measures the software's quality in a whole. In 1976, Boehm and his student raised an idea of software quality quantitative evaluation along with over 60 relevant equations for measuring software product qualities which not only explained how to use hierarchical model to evaluate a software product quality, but also claimed how to use hierarchical relationship model to evaluate the software product quality. Besides, by introducing numerical measuring methods focusing on the lowest layer metric units, they can obtain an overall evaluation method for the software product quality. However, we find this evaluation method mainly focus on the usability, maintainability and portability this three aspects [1-7].

In 1978, McCall and Walters claimed to evaluate software quality by using hierarchical method as well. This includes a hierarchical software quality model consists of measurement, criterion and elements. The criterion mainly focuses on the operation, modification and transformation. They also concluded 11 quality elements for this hierarchical model and pointed relationships between each quality elements. They considered the direct reflect factor of the quality of software products is element. The rule for evaluating software product was only the attribute method. By evaluating these software product's attributes numerically, general condition of software product's quality could be fully shown. This makes it difficult to set all software products' quality element to the best. Evaluation elements should be formulated based on different type or application. So, in the real process of evaluating software products, we should select

ISSN: 1975-0080 IJMUE

different elements to evaluate different product's quality based on different type and application.

In the year of 1985 and 1993, ISO proposed several reports about the technology for evaluating software product quality and some international standards related to SQM. That is software product quality measurement model.

In 1991, ISO proposed 'Evaluation and Guidelines for Information Technology Software Product Quality', IEC 9126:1991. In the 90s, ISO modified IEC9126:1991 as two relevantly independent but different standard series. One of them is 'Software Quality Evaluation Model' which is an instructional evaluation model relevant to IEC 14598; the other one is the instructional quality model IEC 9126 for 'Software Quality Model'. There is a statement in IEC 9126 says: we must measure all the attributes of software quality in a proper degree to test the fitness of our demand. In this way, we could evaluate software quality when we are in the process of developing or maintaining the software [1-9].

Though the quality of software projects is very important, there haven't occurred a systematical standard and method for quality evaluation until now. Thus, this paper will use some models and standards to explore a quality assessment method which is based on the Web application system. Hope to improve the corresponding software system's quality evaluation and the integral grasp of the product quality. And quality measurement is the most significant factor on deciding success of quality control.

Software quality is defined as the totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied requirements. Each characteristic can be divided in other sub-characteristics [22].

2. Related Work

If we implement only one quality index for evaluating quality, we call this is a one point index problem. If we implement more than one index, then we call this is a multi-index problem. Single index is simple, while multiple indexes will influence each other. Besides, different indexes have different significance during the evaluation which increases the complexity. We come across multi-index problems most in our life. Thus, we have to consider different factors from all aspects when doing evaluation to set a general evaluation standard. The evaluation methods set from different aspects also can be a synthesis of different methods such as FMEA, Fuzzy Comprehensive Evaluation, AHP, Expert Assessment Analytical Method, Grey Relation Theory Evaluation Method, Superior Order Method, etc. In the following part, we will introduce FMEA, Fuzzy Comprehensive Evaluation, AHP, and Superior Order Method briefly [10-11].

2.1. Failure Mode and Effect Analysis

Failure Mode and Effect Analysis (FMEA) was first implemented in 1960 in the mission of Apollo in aircraft industry and was set as a military standard by the U.S army in the 80s. It is a systematical tool for industrial design which uses charts to help engineers to evaluate and also is used in engineering analysis. FMEA aims at improving reliability of software products and software product manufacturing, It points out the reliability of software product can be improved during the design period and, in this way, improve the quality of the software product, reduces the product's cost losses.

2.2. Fuzzy Comprehensive Evaluation

Fuzzy Comprehensive Evaluation is a method based on Fuzzy mathematical model. This method turns qualitative evaluation to quantitative evaluation on the basis of mathematical membership theory, which is to evaluate the general of the things, object or products that are confined by several factors by using Fuzzy mathematics. It has a clear evaluation result, a strong system and can solve problems that are fuzzy and difficult to quantization. It is suitable for evaluating all kinds of uncertain problems.

2.3. Analytic Hierarchy Process

Evaluation of analytic hierarchy process (AHP) is a kind of flexible, easy to analyze method which is used often. It is also a useful method in evaluating quantitatively for qualitative problems. Another attribute of AHP is that it makes each factor systematic by separating them orderly and hierarchically. AHP has a method for determining quality elements. This method combines subjective and objective quality factors effectively, compares each two quality factors, and evaluates the importance of factors quantitatively then obtain the relatively importance order as well as the corresponding weight. We normalize the weight then get the weight can be used for evaluation.

2.4. Optimum Order Method

Optimum Order Method sorts evaluation indexes and setting evaluation points for each number, calculates the total superior order number of the evaluation object by synthesizing all evaluating indexes, and then obtain the good and bad order according to the total superior order number.

Optimum Order Method compares each two multi-objective problems then gives a superior order for all the projects. It is simple to use and can be used to solve both quantitative and qualitative problems. Based on management achievement system, we set an evaluation method by using optimum order method.

3. A New software Quality evaluation Model for Web Application

Based on the quality attributes of Web application system, we summarize the weights by combining the application process emphasis and set up a quality evaluation model as show in Table 1. Before measuring the system quality quantitatively, we must obtain the weight of each first quality attributes, second quality attributes and metric units [12-15].

Table 1. A New Quality Evaluation Model

Quality Attributes	First weight	Second weight	Quality attribute metric
Functionality R_1	（W_1）	Functional Defects （w_{11}）	（r_{11}）
		Functional Disabled （w_{12}）	（r_{12}）
		Unconformity of design and implementation （w_{13}）	（r_{13}）
		Design problems （w_{14}）	（r_{14}）
Reliability R_2	（W_2）	Reliability compliable problem （w_{21}）	（r_{21}）
		Functional problem （w_{22}）	（r_{22}）
Usability R_3	（W_3）	Intelligibility problem （w_{31}）	（r_{31}）
		Usability problem （w_{32}）	（r_{32}）
		Operational problem （w_{33}）	（r_{33}）
Hardware usability R_4	（W_4）	Hardware problem （w_{41}）	（r_{41}）

Where In Table 1,

R_i : Represents quality attributes metrics; comes from the sum of product of sub-attribute and weight.

W_i : Represents the weight of quality attributes.

r_{ij} : Represents sub-quality attribute metrics; comes from testing.

w_{ij} : Represents the weight of sub- quality attribute.

When allocating weights for metric units and quality attributes, they imply a proportional relationship between each elements in the set of total weight of 1. If not all the elements are selected when measuring, then the weight will be reallocated. The weight will be allocated to the selected elements, that is the weight of non-selected elements will be 0 [16-19].

3.1. The Weighting Algorism for Metric Values

Generally speaking, project bug is divided into 4 levels. Each level influences product quality differently, thus it is important to evaluate weight for different level's bug accurately. The weighting method is shown in Table 2.

Table 2. Weighting Methods for Handing Different Level Bugs

Bug level	Definition	Influence on the system	Weighting equation
Urgent	There is error in the main module of system or the whole system, and no other temporary way to bypass the error.	The problem has a significant effect on testing progress.	$y_1 = m_1(1 + \log_{10} x)$
High	Major modules of the system occur errors, but there have a temporary solution to bypass the existing problems.	Test work can be implemented,. Problem has a great influence on the progress of the test.	$y_2 = m_2(1 + \frac{1}{2}\log_{10} x)$
Medium	The secondary function or document occur error.	Test work can continue; the problem has a smaller effect on testing progress.	$y_3 = m_3$
Low	Program or document is necessary to be improved and perfection.	Test work can go on ; the problem has no effect on test schedule.	$y_4 = m_4 \frac{1}{\log_{10} x + 1}$

In Table 2, y_i is Corresponding level's equivalent Bug numbers, m_i is Corresponding level's Bug numbers, and x is Corresponding function's test case numbers.

Measurement formula of what we need have to meet the requirements of the above and also to meet the metric is in the interval [0, 1]. When the defect number of all levels is 0, the corresponding y_i in the table also is zero. So the metric formula is as in Eq. (1).

$$w_j = 1 - \frac{\sum_1^4 y_i}{x + \sum \ (m_1 * \log_{10} x + m_2 * \frac{1}{2} \log_{10} x + m_3 + m_4 * \frac{1}{\log_{10} x + 1})} \tag{1}$$

Where w_j represents the metric of j's quality attribute.

The result of the Eq. (1) ensures that all metric units are between the range of 0 ~ 1. 1 represents the measurement is best, while 0 indicates the measurements is the worst. We consider the quality of the product is the better if the metric value more close to 1.

3.2. Fuzzy Mathematical Evaluation Method

Software reliability evaluation is affected by many factors. If we set the factors into the secondary Fuzzy comprehensive evaluation, we can avoid the disadvantage of single stage evaluation without making evaluation too complicated [20-21].

(1) We need to determine a set of the first-level quality attribute metric R, a set of secondary quality attribute metric r and a set of first-level quality weight W, a set of secondary quality metric weighting w, respectively when doing Fuzzy comprehensive evaluation. Comprehensive evaluation's calculation model is the

set of elements. Among them, R_i is an element of metric R, indirectly from the secondary metric. Wi is the element of W. w_{ij} is the element of w. It comes from 7 + 2 decision table. r_{ij} is the elements of the r from the test data after dealt with the weighted.

(2) Use AHP method to determine each element's weighting since it is the measurement of the influence of reliability and is one of the key to Fuzzy judgment. Weighting value can be determined by several methods such as system theories hierarchical analyze method, grey relational analysis method and scoring by experts etc. In this paper, we apply AHP hierarchical analysis method to Set up the system's hierarchical structure. We have the steps as follows:

Step 1: Use 1-9 scale method to compare each two elements to build a comparative judgment matrix of any two elements.

Step 2: Check the judgment matrix's consistency and make necessary adjustments.

Step 3: Calculate each influence element's weighting by focusing on a single standard.

Step 4: Calculate sorting weighting of the current level's element regards to the total target.

Step 5: Normalize the calculated standard weighting vectors respectively.

Among which, step 2's value comes from the 7+2 judgment chart based on the analysis on people's ability to distinguish information levels made by psychologists. Set up the judgment matrix W by comparing the importance of each two level two's w_{ij} in Table 1. As for W=$(a_{ij})_{m*m}$, m is the number of elements.

a_{ij}=1 :element i is as important as element j regarding to the upper level's elements.

a_{ij}=3: element i is a little more important than element j

a_{ij}=5 :element i is more important than element j

a_{ij}=7 :element i is much important than element j.

a_{ij}=9: element i is be of prime importance comparing to element j.

a_{ij}=2n: the importance of element i and element j is between a_{ij}= 2n- 1 and a_{ij}= 2n+1, where n= 1 , 2 , 3 , 4

$a_{ij}=\frac{1}{n}$, if and only if a_{ij}=n, where n= 1 , 2 , …… , 9

The calculation method of steps 4 and 5 above is: solve the judgment matrix eigenvalue, and then calculate the biggest true eigenvalue λ_{max}, and then find the eigenvectors corresponding to its W, then normalize the corresponding feature vector, we can get the corresponding weight vector. The secondary factors weights w_{ij} can be obtained according to the above steps, primary factor weight W_i can be obtained in the same way.

We can obtain the sub-measurement matrix and list each sub-characteristic measurement vector r_i by using the same method.

r_1= (r_{11}, r_{12}, r_{13})
r_2= (r_{21}, r_{22}, r_{23})
r_3= (r_{31}, r_{32}, r_{33})
r_4= (r_{41}, r_{42}, r_{43})
r_5= (r_{51}, r_{52}, r_{53})
r_6= (r_{61}, r_{62}, r_{63})

Supplement the above vectors to a 1*4 vector and obtain an improved normalized r_i.

r_1= (r_{11}, r_{12}, r_{13}, 0)

r_2= (r_{21}, r_{22}, r_{23}, 0)
r_3= (r_{31}, r_{32}, r_{33}, 0)
r_4= (r_{41}, r_{42}, r_{43}, 0)
r_5= (r_{51}, r_{52}, r_{53}, r_{54})
r_6= (r_{61}, r_{62}, r_{63}, 0)

Turn this matrix into corresponding matrix r, presented in Eq. (2).

$$r = \begin{vmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ r_{41} & r_{42} & r_{43} & 0 \\ r_{51} & r_{52} & r_{53} & r_{54} \\ r_{61} & r_{62} & r_{63} & 0 \end{vmatrix} \qquad (2)$$

We can obtain each sub-attribute weighting vector matrix w of evaluation model form 7+2 judgment chart, presented in Eq.(3).

$$w = \begin{vmatrix} w_{11} & w_{12} & w_{13} & 0 \\ w_{21} & w_{22} & w_{23} & 0 \\ w_{31} & w_{32} & w_{33} & 0 \\ w_{41} & w_{42} & w_{43} & 0 \\ w_{51} & w_{52} & w_{53} & w_{54} \\ w_{61} & w_{62} & w_{63} & 0 \end{vmatrix} \qquad (3)$$

Then, we can obtain the first -level characteristic weighting value vector from 7+2 judgment Table , as follow:

$$W = (W_1,\ W_2,\ W_3,\ W_4,\ W_5,\ W_6)$$

The first-level measurement vector R= (R_1, R_2, ……, R_i) is the sum of the product of the second-level weight and its metric value.

Suppose Matrix M= w× r^T 's m_{ii} (m_{11}, m_{22}, m_{33}, m_{44}) form the first measurement vector that is R_i=m_{ii}, thus:

$$M= w \times r^T \qquad (4)$$

$$R= (m_{11},\ m_{22},\ m_{33},\ m_{44})$$

Obtain the evaluation value S, presented in Eq.(5).

$$S= (R \times W^T) \qquad (5)$$

Multiply the equation above by 100, we obtain the final quality evaluation value calculation formula, presented in Eq.(6).

$$Si= S*100 \qquad (6)$$

4. Experimental Results

In this section we conclude test data of three Web applications to evaluate the project quality. The data we need to make evaluation includes number of test cases and the number of different level's defects corresponding to the second level's attributes.

The experiment includes 1465 pieces of test cases which are distributed as follows: the number of functional case is 787, the number of reliability case is 353, the number of usability case is 196, the number of standard is 129. See details in Table 3.

Table 3. Test Cases

Quality attributes	Sub-attributes	The number of Test cases
Functionality	Function defects	132
	Function disabled	155
	Unconformity between design and implementation	165
	Design problems	335
Reliability	Reliability compliance problem	298
	Performance problem	55
Usability	Intelligibility problem	67
	Learning problem	61
	Operating problem	68
Hardware standard	Hardware problem	129

There are 132 pieces of defects in all and the distribution according to four function areas are as follows: functionality 71, reliability 45, usability 5, standard 11. See details in Table 4.

Table 4. Summary of Test Result

Quality attributes	Sub-attribute	Total	Urgent	High	Medium	Low
Functionality	Function defects	9	2	6	1	0
	Function disabled	14	3	10	1	0
	Unconformity between design and implementation	5	0	1	4	0
	Design problem	43	5	20	17	1
Reliability	Compliance problem	42	17	17	8	0
	Performance problem	3	0	2	1	0
Usability	Intelligibility problem	2	0	1	1	0
	Learning problem	2	0	0	2	0
	Operational problem	1	0	0	0	1
Hardware standard	Hardware problem	11	1	9	1	0

4.1. Weighting Determination of the First- level Attributes

System’s first-level quality attributes include functionality, reliability, hardware standard, usability. Functionality means the degree its function meets the design standard and the level it satisfy its customer’s need. Reliability means the degree the product could maintain its performance in a certain time period and a certain environment. Hardware standard means the ability of its hardware to complete corresponding product’s function. Usability means the effort the customer put into learning, manipulating, preparing for inputs and comprehending the output of the software. For Web application, it is a must to finish all the required functions without error. Then all the functions should be reliable. And then it has to make sure that hardware will not influence the implementation of system’s function. At last, it should make the system as easy as possible to learn, understand and manipulate. Thus, functionality is more important than reliability. Reliability is more important than hardware standard. Hardware standard is more important than usability. And functionality is also more important than hardware standard. Reliability is more important than usability. In the similar way, we can obtain the relationship between each functional attributes. Their relationships can be illustrated by AHP method and 7+2 judgment tablet. That is the judgment matrix, shown in Table 5.

Table 5. Judgment Matrix for Weighting Value of the First-level Attributes

	Functionality	Reliability	Hardware standard	Usability
Functionality	1	2	4	6
Reliability	1/2	1	2	4
Hardware standard	1/4	1/2	1	2
Usability	1/6	1/4	1/2	1

We have $\lambda_{\max}$=4. 0104,

CI=($\lambda_{\max}$-n)/(n-1) = (4. 0104-4)/(4-1) =0.00347。

RI = 0.89。

CR=CI/RI =0.00347/0.89= 0.039< 0.1。

Thus, by using AHP judgment rules, we find the judgment matrix satisfy consistency and no need to modulate. By using table 5, we can calculate the eigenvector which is （0.8513， 0.4563， 0.2281， 0.1229） . In normalized performance W= (0.5133, 0.2751, 0.1375, 0.0741).

4.2. The Weighting Determination of the Second -level's Characteristics

1) The quality attributes of function defect of functionality:

For the first experiment project, Table 6 shows the summary of the defect of severity level for functionality.

Table 6. The Summary of the Test Defects for Severity Level

Total（x）	Urgent(m_1)	High(m_2)	Medium(m_3)	Low(m_4)
132	2	6	1	0

Functional sufficiency of metric units:
Definition of metric, presented in Eq.(7).

$$r_{11} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1 * \log_{10} x + m_2 * \frac{1}{2} \log_{10} x + m_3 + m_4 * \frac{1}{\log_{10} x+1}} = 0.871 \qquad (7)$$

In the same way we can obtain other metric value of relevant second-level quality attributes by using weighting formula.

2) The quality attributes of function disability of functionality, presented in Eq.(8).

$$r_{12} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1 * \log_{10} x + m_2 * \frac{1}{2} \log_{10} x + m_3 + m_4 * \frac{1}{\log_{10} x+1}} = 0.831 \qquad (8)$$

3) The quality attributes of unconformity between design and implementation of functionality, presented in Eq.(9).

$$r_{13} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1 * \log_{10} x + m_2 * \frac{1}{2} \log_{10} x + m_3 + m_4 * \frac{1}{\log_{10} x+1}} = 0.970 \qquad (9)$$

4) The quality attributes of design problem of functionality, presented in Eq.(10).

$$r_{21} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1 * \log_{10} x + m_2 * \frac{1}{2} \log_{10} x + m_3 + m_4 \frac{1}{\log_{10} x+1}} = 0.811 \qquad (10)$$

5) The quality attributes of reliability compliance problem of reliability, presented in Eq.(11).

$$r_{22} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.739 \qquad (11)$$

6) The quality attributes of performance problem of reliability, presented in Eq.(12).

$$r_{23} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.921 \qquad (12)$$

7) The quality attributes of intelligibility of usability, presented in Eq.(13).

$$r_{31} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.932 \qquad (13)$$

8) The quality attributes of learning of usability, presented in Eq.(14).

$$r_{32} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.968 \qquad (14)$$

9) The quality attributes of manipulation of usability, presented in Eq.(15).

$$r_{33} = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.995 \qquad (15)$$

10) The quality attributes of hardware standard, presented in Eq.(16).

$$r_4 = 1 - \frac{\sum_1^4 y_i}{x+\sum m_1*\log_{10}x+m_2*\frac{1}{2}\log_{10}x+m_3+m_4*\frac{1}{\log_{10}x+1}}=0.851 \qquad (16)$$

Among which:y_i represents the number of tests cases, by which we find defects during testing. x=the number of functions of evaluation.

The more r_i is close to 1 the better result will be. The more r_i is close to 0 the worse result will be.

4.3. Quality Evaluation of Measuring Matrix

Now, the corresponding measuring matrix of the three evaluated web projects about the given input value is as follows.

1) Measuring matrix r1 of the first Web project, presented, presented in Eq.(17).

$$\text{r1} = \begin{vmatrix} 0.871 & 0.831 & 0.970 & 0.811 \\ 0.739 & 0.921 & 0 & 0 \\ 0.932 & 0.968 & 0.995 & 0 \\ 0.851 & 0 & 0 & 0 \end{vmatrix} \qquad (17)$$

In the same way, we have other two corresponding measuring matrix about the evaluated project as follows:

2) Measuring matrix r2 of the second Web project, presented in Eq.(18).

$$\text{r2} = \begin{vmatrix} 0.746 & 0.906 & 0.943 & 0.833 \\ 0.908 & 0.973 & 0 & 0 \\ 0.950 & 0.919 & 0.999 & 0 \\ 0.850 & 0 & 0 & 0 \end{vmatrix} \qquad (18)$$

3) Measuring matrix r3 of the third Web project, presented in Eq.(19).

$$\text{r3} = \begin{vmatrix} 0.782 & 0.834 & 0.924 & 0.964 \\ 0.882 & 0.905 & 0 & 0 \\ 0.905 & 0.857 & 0.898 & 0 \\ 0.946 & 0 & 0 & 0 \end{vmatrix} \qquad (19)$$

Measuring value R1 of first- level quality attribute can be obtained from Eq. (4).

M= r1×$w1^T$

$$=\begin{vmatrix} 0.871 & 0.831 & 0.970 & 0.811 \\ 0.739 & 0.921 & 0 & 0 \\ 0.932 & 0.968 & 0.995 & 0 \\ 0.851 & 0 & 0 & 0 \end{vmatrix} \times \begin{vmatrix} 0.5650 & 0.2622 & 0.0553 & 0.1175 \\ 0.75 & 0.25 & 0 & 0 \\ 0.3089 & 0.1096 & 0.5815 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}^T$$

$$=\begin{vmatrix} 0.8589 & 0.6590 & 0.8354 & 0.4808 \\ 0.861 & 0.7845 & 0.941 & 0.6382 \\ 0.9242 & 0.3292 & 0.9726 & 0.2629 \\ 0.871 & 0.739 & 0.932 & 0.851 \end{vmatrix}$$

Thus R1={0.8589, 0.7845, 0.9726, 0.851}. That is the first level measuring matrix of the first Web project. Table 7 .

Table 7. Summary Result of Weightings

Quality attributes	First-level weighting	Quality sub-characteristics	Second-level weighting
Functionality	0.5133（W_1）	Function defect	0.5650（w_{11}）
		Function disabled	0.2622（w_{12}）
		Unconformity between design and implement	0.1175（w_{13}）
		Design problem	0.0553（w_{14}）
Reliability	0.2751（W_2）	Reliability compliance problem	0.7500（w_{21}）
		Performance problem	0.2500（w_{22}）
Usability	0.0741（W_3）	Intelligibility problem	0.3089（w_{31}）
		Learning problem	0.1096（w_{32}）
		Manipulation problem	0.5815（w_{33}）
Hardware applicability	0.1375（W_4）	Hardware problem	1（w_{41}）

By using data weighting method and the method of corresponding first-level measurement value, we obtain the final measurement table of the first project, shown in Table 8.

Table 8. Quality Evaluation Measurement Table

Quality attributes	First-level measurement value	Quality sub-characteristics	Second-level measurement value
Functionality	0.8589（R_1）	Function defect	0.871（r_{11}）
		Function disabled	0.831（r_{12}）
		Unconformity between design and implementation	0.970（r_{13}）
		Design problems	0.811（r_{14}）
Reliability	0.7845（R_2）	Reliability compliance problem	0.739（r_{21}）
		Function problem	0.921（r_{22}）
Usability	0.9726（R_3）	Intelligibility problem	0.932（r_{31}）
		Learning problem	0.968（r_{32}）
		Manipulation problem	0.995（r_{33}）
Hardware applicability	0.851（R_4）	Hardware problem	0.851（r_{41}）

By calculating weighting matrix and measurement matrix, we obtain the final evaluation value from Eq.(5): S=R1×W^{T}. Since S1=R1×W^{T}=0.8589 0.7845 0.9726 0.851×$\begin{matrix}0.5133\\0.2751\\0.0741\\0.1357\end{matrix}$=0.8458, we have the evaluation value of the first Web project S1=0.8458.

In the similar way, we can obtain evaluation value of two other projects:
S2=0.8586 for the second Web project.
S3=0.864 for the third Web project.

Multiply the above equation by 100, we have the formula for calculating final quality evaluation value, presented in Eq.(20).

$$S1=S*100 \tag{20}$$

And the evaluation value in centesimal:

- S11= S*100=84.58 for the first project.
- S21=85.86 for the second project.
- S31=86.4 for the third project.

5. Conclusions

This paper discusses the software quality evaluation model. It combines the quality management concept of Web application and proposes an improved quantitative quality measurement model. This model applies the AHP and Fuzzy Mathematical method. This paper also discusses the selection of Web application system models and the evaluation methods. It gives the evaluation method and corresponding formulas suitable for Web application project. Finally, it proposes a more targeted weighting algorism for evaluation model and measurement value.

In the future, we will continue to do experiments and investigate other characteristics and sub-characteristics of quality model. We will define more accurate weight value for the quality characteristics and sub-characteristics by the influence between the quality characteristics and sub-characteristics.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61170273).

References

[1] B. W. Boehm, J. Brown and H. Kaspar, "Characteristics of Software Quality [A]", TRW Serious of Software Technology [M], New York: NorthHolland, vol. 1, (**1978).**

[2] ISO/IEC9126, Information Technology—Software Product Evaluation—Quality Model[S], (**2001).**

[3] R. G. Dromey, "Cornering the Chimera [J]", IEEE Software, vol. 13, no. 1, (**1996**), pp. 33243.

[4] ISO9126, Information Technology Software Product Evaluation Quality Characteristics and Guidelines for Their Use [S], International Organisation for Standardization, (**1992**).

[5] J. McCall, P. Richards and G. Walters, "Factors in Software Quality", [R], Technical Report CDRL A003, U S Rome Air Development Centre, 1977.G. Karner. Resource Estimation for Objectory Projects. Objective Systems, vol. 1, (**1993**).

[6] H. Ling, J. Ma and F. Y. Zou, "Software Dependability Evaluation Model Based on Fuzzy Theory [A], International Conference on Computer Science and Information Technology, (**2008).**

[7] T. L. Saaty, "The Analytic Hierarchy Process [M], New York: McGraw Hill, (**1980).**

[8] P. J. Boland and H. Singh, "A birth-process approach to Moranda's geometric software reliability model [J]", IEEE Trans. on Reliability, vol. 52, no. 2, (**2003**), pp. 168-172.

[9] M. A. Stutzke and C. S. Smidtz, "A stochastic model of fault introduction & removal during software development [J]", IEEE Trans. on Reliability, vol. 50, no. 2, (**2001**), pp. 184-193.

[10] P. Popov, L. Strigini and S. Kuball, “Estimating bounds on the reliability of diverse systems [J]”, IEEE Trans. on Software Engineering, vol. 29, no. 4, (**2003**), pp. 345-359.
[11] O. Korkmaz, I. Akman, and S. Ostrovska, “Assessing Software Quality Using the Markov Decision Processes”, Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 24, no. 1, (**2014**), pp. 86–104.
[12] E. S. Andersen and S. A. Jessen, “Project Evaluation Scheme: A tool for evaluating project status and predicting project results”, Project Management vol. 6, no. 1, (**2000**), pp. 61–9.
[13] R. Yu and G. H. T. Zeng, “A soft computing method for multicriteria decision making with dependence and feedback”, Applied Mathematics and Computation, vol. 180, (**2006),** pp. 63–75.
[14] T. L. Saaty, “Decision Making with Dependence and Feedback: The Analytic Network Process”, RWS Publications, Pittsburgh, (**2001).**
[15] A. Alami, B. Wong, and T. McBride, “Relationship Issues in Global Software Development Enterprises”, Journal of Global Information Technology Management, vol. 11, no. 1, (**2008).**
[16] Y. Jiangl, Y.-N. Li, Y. Du and X.-D. Fu, “The Multi-Dimension Component Quality Evaluation”, 201O International Conference on Computer Design and Applications (ICCDA 2010).
[17] K. M. Oliveira, A. R. Rocha, and K. C. Weber, Workshop on software quality, ICSE 02, May 2002, USA, pp. 671-672.
[18] J. A. McCall, P. K. Richards and G. F. Walters, "Factors in Software Quality," Preliminary handbook on software quality for an acquisition manager, vol. 3, Technical Report RADC-TR-77-369, Hanscom AFB, MA OI731, (**1977**).
[19] ISO 9126, "Information Technology - Product Quality - Part!: Quality Model ", International Standard ISO/IEC 9126, International Standard Organization, (**2001**), June.
[20] W.-J. Han and T.-B. Lu, “Study On Quality Evaluation Model of Communication System.
[21] A. Alvaro, E. S. Almeida and S. R. 1. Meira, "Towards a Software Component Quality Model," the 31st IEEE EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Work in Progress Session, Porto, Portugal. (**2005**).
[22] Y. Jiang, Y.-N. Li, Y. Du and X.-D. Fu, “The Multi-Dimension Component Quality Evaluation, 2010 International Conference on Computer Design and Applications (ICCDA 2010).

Authors

Wan-Jiang Han, she was born in HeiLongJiang province, China, 1967. She received her Bachelor Degree in Computer Science from Hei Long Jiang University in 1989 and her Master Degree in Automation from Harbin Institute of Technology in 1992.

She is an assistant professor in School Of Software Engineering, Beijing University of Posts and Telecommunication, China. Her technical interests include software project management and software process improvement.

Tian-Bo Lu, he was born in Guizhou Province, China, 1977. He received his Master Degree in computer science from Wuhan University in 2003 and his PH.D Degree in computer science from the Institute of Computing Technology of the Chinese Academy of Sciences in 2006. He is an Associate professor in School of Software Engineering, Beijing University of Posts and Telecommunications, China. His technical interests include information and network security, trusted software and P2P computing.

Xiao-Yan Zhang, she was born in Shandong Province, China, 1973. She received her Master Degree in Computer Application in 1997 and her PH.D Degree in Communication and information system from Beijing University of Posts and Telecommunication, China, in 2011.

She is an Associate professor in School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her technical interests include software cost estimation and software process improvement.