International Journal of u- and e- Service, Science and Technology
Vol.10, No. 1 (2017), pp.89-104
http://dx.doi.org/10.14257/ijunesst.2017.10.1.08

Research on Software Trustworthiness Evaluation for Web Application Based on Software Product

Wanjiang HAN[1], Suyi LI [2], Haoran JIA[2], Tianbo LU[1], Jincui YANG[1], Ruotong YU[1] and Yang LI[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

Due to the different needs of the different software projects of trustworthiness evaluation, it is a difficult research topic to build a unified quantitative trustworthy software evaluation model. This paper puts forward a kind of trustworthiness evaluation method, then quantitative software trustworthiness evaluation model is proposed on the analysis of trustworthiness needs of Web based software project on whole life cycle. Firstly, a structural model of trustworthiness index tree is designed based on software product driven. Then AHP fuzzy algorithm is used to analyze the weights of the model. Finally, the analysis and experimental results of this paper show that the proposed model is effective and rational. This software trustworthiness evaluation model has a certain practical value for Web Application.

Keywords*: software trustworthiness; software trustworthiness evaluation model; Web application evaluation; Based on software product evaluation; AHP fuzzy algorithm*

1. Introduction

The software trustworthiness, as a new (quality) measurement of the software, has been playing an important role in the software engineering area and it is inevitable to construct trustworthy software during the development of the modern software engineering technologies. Along with the increasing scale of the software, the job to develop, integrate and maintain a product is becoming more and more complex. However, the paucity of the trustworthy software construction and the paucity of the trustworthiness measurement and evaluation indicate that the software products are born with some flaws, which constitutes a threat to the software system [1-3].

The trustworthiness description of ISO/IEC 15408 standard [4]: A trustworthy behavior of component, operation or process is predictable under any kinds of operation circumstance with a well-done defense mechanism to keep the destruction brought by application software, virus or the physical interference. Trusted computing organization [5] thinks that an entity is trustworthy only if it operates in the expected way according to the targeted aim. Algirdas and his fellows [6] argue that the trustworthiness of the traditional software consists of two aspects, namely security and reliability.

Currently, the trustworthiness evaluation standard is suffering from the lack of quantitative information because most evaluations are qualitative. Besides, many evaluations are directly aimed at the final software product. This paper aims at the Web applications, building a trustworthiness evaluation model based on the attributes of the software product. The built model in this paper is a quantitative model. Through the evaluation of the trustworthiness attribute, we can get the quantitative trustworthy results from the Web applications systems intuitively and credibly.

2. Related Work

The fundamental evaluation model of the software trustworthiness is shown as Figure 1. First, determine trustworthiness properties, the related property relations and the weight. According to the relevant evidence obtained by properties to assess the credibility degree of the software project. Mainly, there are two kinds of comprehensive assessments of the software trustworthiness: product-oriented assessment and process-oriented assessment. The former one assess the credibility of a software based on the results obtained by analyzing and testing software products. The latter one assess the credibility through analyzing whether the development process obeys the trustworthiness convention during the whole lifetime of a software [9].

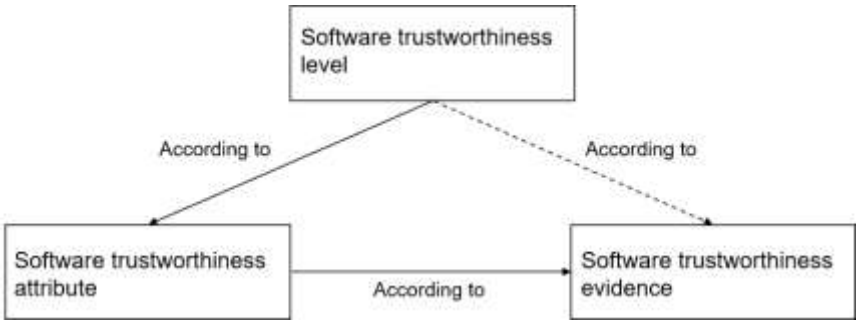

Figure 1. The Basic Model of the Software Trustworthiness

2.1. Trustworthiness Research based on Product

The software products not only compromise source code and executable program, but also consist of the submitted documents which come out during the development process. Therefore, it is necessary for the software product-oriented assessment to evaluate all of them, including source code, fundamental functions and other apropos things in the product [9-12].

Based on the theory of software trustworthiness evaluation, we need two steps to achieve our goal. Firstly, researchers should analyze the demand of the software and acquire the trustworthiness attribute on which the software focus. Secondly, researchers should analyze and test the products with the help of evaluation tools and evaluation methods. After this, they will obtain a large number of useful information and based on these information, they are able to assess the trustworthiness of a software [9].

2.2. Trustworthiness Research based on Process

Amoroso and his fellows [9-13] put forward a process-oriented trustworthiness assessment, which determines the final trustworthiness of a software by assessing the trustworthiness in different stages during the whole lifetime of a software. They firstly enact the trustworthiness principles and then analyze whether there is a violation of the principles during the development process to assess the software trustworthiness. The merits of this assessment is that it keeps the subjective factors away by assessing the matched-degree with the principles. Yet, there are also some questions when the assessment is applied.

2.3. Analysis the Weights of the Trustworthiness Attribute

There are many ways to determine the weights, for instance, AHP and fuzzy algorithm *etc.*, The followings will illustrate AHP method and how to evaluate the trustworthiness weights by triangular fuzzy algorithm based on AHP method.

AHP method was developed by Doctor Thomas L. Saaty and it is accorded with the multi-criteria complex decision. AHP divides the complex decision situations into several parts and then organizes these parts to construct tree hierarchical structure. After this, we distribute weights to each parts according to their relative importance. Finally, we should analyze the priority of each parts. The steps of the AHP hierarchical analysis can be seen in Figure 2.

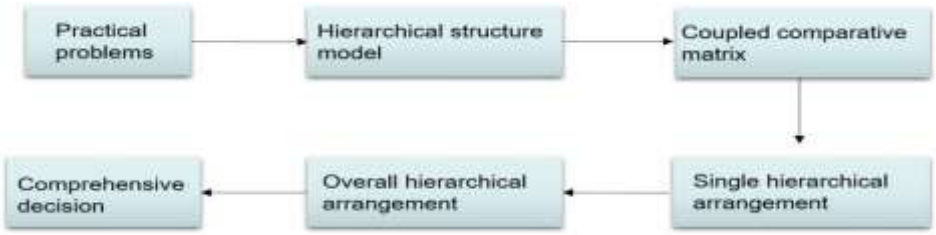

Figure 2. The Steps of the AHP Hierarchical Analysis

1. Construction of the hierarchical structure model

A good hierarchical structure is the key to solving problems. Good hierarchical structure is based on the decision makers' comprehensive knowledge of the problems: what the hierarchical analysis concerns is relative weight between the lowest structure and the highest structure. The relative weight serves as the cornerstone for arranging the order of the various kinds of plans and methods in the lowest structure by which the leader can make the final decision.

2. Construction of the comparative (judgement) matrix

After the construction of hierarchical structure model, the relationship of subordination between two layers has been decided. Suppose the element C_k in the upper layer serves as principle, which can dominates the elements $A_1,A_2,A_3,...A_n$ in the lower layer. What we want to do then is to distribute relevant weights to each element on the basis of their relative importance under the control of the principle C_k. The qualitative results of the weight distribution between each layers are usually unconvinced. Therefore, Saaty and his fellows came up with an idea of building coupled comparative matrix $A=[a_{ij}]_{n*n}$, where a_{ij} is the comparative result of comparing ith and jth elements. We do not compare all the elements together in a comparative matrix, however, we compare each pair of elements according to the relative scalar. In this way, it will be much easier while comparing elements of different attributes and meanwhile, of more accuracy.

3. Single hierarchical arrangement

The purpose of the single hierarchical arrangement is to find how much impact the elements in the lower layer will exert on a certain element in the upper layer. And this can be done by finding the characteristic root and characteristic vector of the comparative matrix, the coupled comparative matrix A. What we need to find are characteristic root and characteristic vector of $AW=\lambda_{max}\mathbb{W}$ where λ_{max} is the biggest characteristic root of A and W is the corresponding characteristic vector, the weight of the elements in the single hierarchical arrangement. The most popular way to calculate the biggest characteristic root and characteristic vector is harmonization method.

Normalize each column in the coupled comparative matrix: $\bar{a} = \frac{a_{ij}}{\sum_{k-1}^{n} a_{kj}}, i, j \in 1,2,3.....,n$

Add up each row of the normalization matrix

Normalize the vector $W_i = \frac{M_i}{\sum_{j=1}^{n} M_j} i \in 1,2,3....n$

$M=(M_1,M_2,....,M_n)^T$: $M_i = \sum_{j=1}^{n} \overline{a_{ij}}, i \in 1,2....n$, and we will get $W=(W_1,W_2,...,W_n)^T$, which is the characteristic vector;

Calculate the biggest characteristic root of the coupled comparative matrix: $\lambda_{\max} = \sum_{i=1}^{n} \frac{(AW)}{nW_1}$

where $(AW)_i$ is the ith element in the vector AW.

4. Consistency check of single hierarchical arrangement

We must check the consistency of the comparative matrix when we are going to construct a comparative matrix. Consistency check is based on the matrix theory. According to the matrix theory, we have the formula AW= $\lambda_{\max} W$, especially $\lambda_{\max} = n$ (n is the order of A) when the coupled comparative matrix have the same consistency. For the purpose of checking the consistency of coupled comparative matrix, it is necessary to calculate $(\text{Consistency Index, CI}) = \frac{\lambda_{\max} - n}{n-1}$ the larger value CI has, the less consistent the coupled comparative matrix are; the smaller value CI has, the more consistent the coupled comparative matrix are.

Due to the difficulties of realizing total consistency, we should at least guarantee satisfactory consistency. For multiple-order coupled comparative matrix, we have the

Consistency Ratio $(Consistency\ Ratio, CR) = \frac{CI}{RI}$ which is the ratio of CI and same-order mean random consistency index(Random Index). When CR<0.1, we consider the inconsistency degree of A is in the range where we can accept, so A has the satisfactory consistency and passes the consistency check. Otherwise, we should construct a new coupled comparative matrix A, and adjust the value of a_{ij}.

5. Overall hierarchical arrangement and its consistency check.

The purpose of the overall hierarchical arrangement is to calculate the relative importance weights of all the elements in a certain layer compared to the highest layer（ overall aim）. And the calculation is done from the highest layer to the lowest layer, layer by layer.

Set A level with m amount of elements $A_1,A_2,A_3.....A_m$.The sequence for total expectation Z shall be $a_1,a_2,....,a_m$, The n amount of elements of B level match the single sequence A_j of A level is $b_{1j},b_{2j},...,b_{nj}(j \in 1,2,...,m)$.So B level's total $\sum_{j=1}^{m} a_j b_{ij}$ sequence shall be （affect sum）;

Set B level with elements $B_1,B_2,...,B_n$. The Single-level sorting Consistency Index for upper layer's(A level) elements $A_j(j \in 1,2,...,m)$ is CI_j,The Random Consistency indicators is

RI_j.So the consistency ratio of total arrangement of hierarchy is $CR = \frac{a_1 CI_1 + a_2 CI_2 + ... + a_m CI_m}{a_1 RI_1 + a_2 RI_2 + ... + a_m RI_m}$

When we think the total arrangement of hierarchy were tested for consistency and the consistency is very well. Otherwise, we need to readjust the value of elements in judgement matrix with high Consistency ratio.

3. Trustworthiness Model of Web Application based on Software Product

Establishing a workable system of quantitative indicators is the basis for a credible assessment software. And different software has different trustworthiness requirement. In order to guarantee the trustworthiness of the Web software, we need to evaluate and monitor the whole software development process. This paper is mainly based on the development of multi-year projects, testing, analysis, evaluation experience, while

referring to the relevant information from the software life-cycle perspective of Web software divides the internal and external, to determine the trustworthiness of the relevant attributes. Determine the attribute weights by AHP and other methods, in order to give trustworthiness assessment model. The following will aim at Web software and show the construction of the trustworthiness evaluating principle system and the evaluating process.

3.1. Trustworthiness Index

Along with the development of the technology of heterogeneous applications interoperability and integration under Internet circumstance, the trustworthiness quality of Web application software receive more and more attention. Trustworthiness evaluation consists of software process evaluation and product evaluation [15]. For the service architecture of Web application, quality factors are more focused on the function evaluation and performance evaluation. One of the evaluations from the users' standpoints is evaluating Web service through clients. Interface and the products are trustworthiness requirements that can be seen by clients. In addition, to ensure the trustworthiness of the development side, it is necessary to evaluate the documents, design, and code of the development side.

Trustworthiness index can be seen as a node in the index tree. And we can define the data structure of the trustworthiness index [16], which can be seen as follows:

```
typedef struct ATNode{ // a node of the index tree
  int   ID ; //attrtibute ID
  string name ;  //name
  boolean  isBaseNode ;//if it is a leaf node
  AofT  *father ; //the parent node
  AofT  *child ;  //the child node
  Float  Weight  //  quantized weight
  int Flag; // Flag =0,1,2 represent root node, intermediate node, leaf node
}
```

Suppose for the attributes of "code", Definetype Code ATNode, then Code.name= Code, Code.Id= 14, Code. boolean=F, Code.father = " the trustworthiness of Web", Code.child ={integrity, normalization, correctness, maintainability, innovative, reusability}。

In order to probe into the trustworthiness of the Web application software, we do research work both internally and externally. The attributes of the main part includes interfaces, documents, codes, design and software product. And interfaces, documents and software product belong to external part while codes and design belong to internal attributes.

As the framework of the collecting the trustworthy evidence, we construct an evaluation model to evaluate different trustworthy software based on that which is much more accordant with the trustworthiness requirements. For the interface, documentation, design, code, software product, corresponds to the first layer attributes, each attribute can decide whether to base node according to the attribute level it contains. The following is hierarchical iterative algorithm of attribute. With the help of the tree structure, we can get the relationship between the trustworthiness requirements and trustworthy index clearly and evaluate the process of the trustworthiness evaluation. The algorithm of generating Trusted attribute tree is shown as following:

Algorithm :
```
ATTree (ATNode *INDB){
ATNode TNode = Head;
Insert( TNode)
{
if( TNode.isBaseNode) break;
for( i = 0; i<INDB. length; i++) {
```

```
if( INDB [i] .fatherID= = FNode. ID) {
FNode.child [j] = INDB [i] .ID;
Insert( INDB [i] ) ;
j++;
}
}
}
ATNode CheckNode = Head ;
Check(CheckNode){
if(CheckNode){
      for(int i =0 ;i <CheckNode .child[ ] .
length -1 ;i ++){
        if(CheckNode .chi ld[ 0] ==null)
          if(!CheckNode .isBaseNode)
return ERROR ;
        Check(CheckNode .chi ld[ i] );}
  }
```

Through the above algorithm, we can generate an index tree with the head node being Head. The function Insert () is used to build an index tree; the function Check () is used to check if the index tree is integral. Both functions are realized with the iterative method. After building an index tree for the evaluated object, it is necessary to regard all the child nodes of each parent node as a subset. Finally, we will determine a trustworthiness index tree which has the hierarchical relationship as shown in Figure 3.

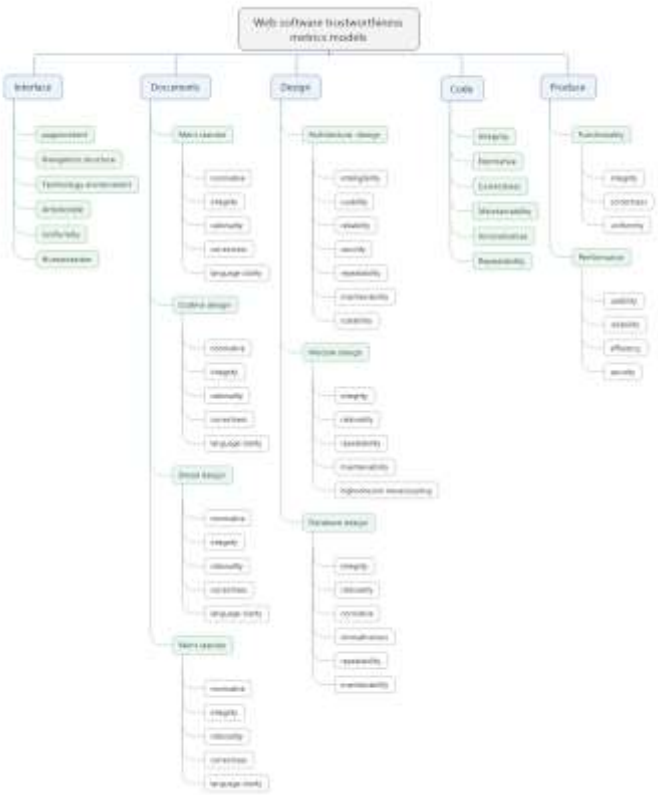

Figure 3. The Hierarchy of the Trustworthiness Attributes

Next, we will introduce index of relevant attributes

3.1.1. The Attribute of Interface: For Web application software, the attributes of the interface evaluation include the page content, navigation structure, technology environment, artistic style, uniformity and humanization. The details are listed in Table 1.

Table 1. Interface Trusted Attribute Description

Index of attributes	Evaluation principle
Page content	● Correct information ● Uniform terms and uniform form of writing, ● Proper arrangement of column and menu ● The correctness of promptness of passing information
Navigation structure	The page must have clear navigational hints which are intelligible and user-friendly
Technology environment	The page size must be appropriate and can be browsed with different kinds of browsers in different resolution ratio; Wrong links and null links are forbidden.
Artistic style	Clear interface Appropriate layout Proper and uniform word size, and proper and uniform typeface Harmonious color style
Uniformity	The style, usage and content of a page should be uniform
Humanization	The pages should be easy and convenient for the users

3.1.2. The Attribute of Documents: The attribute index of the document evaluation includes requirement standard, outline design, detail design, development standard, *etc.*, The trustworthiness of each document consists of normative, integrity, rationality, correctness, language clarity, *etc.*, Therefore, there are two layers attributes included in the "document" node. The Table 2, shows the regulation of the attribute evaluation in detail.

Table 2. Document Trusted Attribute Description

Primary attributes	Secondary attributes	Evaluation principle
Requirement standard	Normative	The document satisfies the given document template
	integrity	The document has covered all the content.
	rationality	The content of the document is logical and the idea is described properly
	correctness	The content of the document is described correctly.
	language clarity	The language in the document is pithy and highly readable.
outline design	Normative	The document satisfies the given document template, including format, sections and language description.
	integrity	The described content in the document is accordant with the requirement without omitting.
	rationality	The content of the document is logical and the idea is described properly
	correctness	The content of the document is described correctly.
	language clarity	The language in the document is pithy and highly readable.
Detail design	Standard ability	The document satisfies the given document template, including format, sections and language description.

	integrity	The described content in the document is accordant with the outline without omitting.
	rationality	The content of the document is logical and the idea is described properly
	correctness	The content of the document is described correctly.
	language clarity	The language in the document is pithy and highly readable.
development standard	Normative	The document satisfies the given document template, including format, sections and language description.
	integrity	The document has covered all the content of the development standard.
	rationality	The content , logic and idea of the document is described properly.
	correctness	The content of the document is described correctly.
	language clarity	The language in the document is pithy and highly readable.

3.1.3. The Attribute of Design: The design part consists of architecture design, module design and database design. Each kind of design has different trustworthiness attribute. Therefore, there are two layers attributes in the "design" node. The attribute evaluation principle is shown in Table 3, in detail.

Table 3. Design Trusted Attribute Description

Primary attribute	**Secondary attribute**	**Evaluation principle**
Architecture design	intelligibility	Refers to the degree of whether the concepts and description of software architecture can be fully expressed and deep understood. Reflects the supporting degree of communicating knowledge, experience and new design between software designers and between designers and users.
	Usability	It refers to the time ratio of the system's normal operating. What may influence this index are the redundant members and fault detection components.
	reliability	Error by the implementation of the software system architecture take place under the system, in the case of accident or misuse, the use of software system architecture strategy to maintain the functional properties of the basic ability.
	security	It refers to the system according to the embodiment of the architecture while providing services to legitimate users can prevent unauthorized users attempt to use or refuse service capabilities.
	repeatability	It refers to the framework of system structure or mapped into the architecture components can be reused or degree of structural member component library architecture library.
	maintainability	According to the system structure, the ability of the system to repair the software system after the error occurred,
	scalability	Architecture of the system should meet the additional requirements of survival during system that allows the addition of new members or structures to meet the new demands of the system capacity.
Module design	integrity	Consistent with software requirements, no missing items.
	rationality	The software module division is satisfied requirements and design principles
	repeatability	The existing software modules and their active ingredients used

		in the construction of new software or system
	maintainability	The ability of the modular system to repair the software system after the error occurred, which the system is implemented.
	high cohesion, loose coupling	Within the software model of high cohesion, low coupling level.
Database design	integrity	Supply the demand, satisfied the overall design of the general requirements, no missing items.
	rationality	Database design basically satisfied the design and demand principles.
	normative	Satisfy the database design standard.
	innovativeness	Whether the database design with ideas of independent.
	repeatability	The existing database structure, tables and their active ingredients used in the construction of new software or system.
	maintainability	The ability of the database system to repair the software system after the error occurred, which the system is implemented.

3.1.4. The Attribute of Code: Code is the internal attribute of the software, including integrity, normative, correctness, maintainability, innovativeness and repeatability, *etc.*, The Table 4 shows the attribute evaluation principle in detail.

Table 4. Description of the Attribute of Code

Attribute index	Evaluation principle
Integrity	Codes are accordant with the requirements and design without omitting items.
Normative	Satisfy the coding standard.
Correctness	No flaws will be reported
Maintainability	The difficulties of modifying codes.
Innovativeness	Whether the codes are innovative
Repeatability	Make the existing codes and their valid components available for building new software and systems.

3.1.5. The Attribute of Product: For software product, we evaluate them mainly through the evaluation of the functions and performance. And there are also some secondary attributes under the functionality and performance. Therefore, there are two layers of attributes included in the "product" node. The Table 5, shows the attribute evaluation principle in detail.

Table 5. Description of the Attribute of Product

Primary attribute	Secondary attribute	Evaluation principle
functionality	integrity	The implementation functions of the software products satisfy the requirements.
	correctness	Correctness refers to the abilities of the software to execute tasks correctly.
	uniformity	The uniformity refers to achieving uniformity of the software functions without ambiguity.
performance	usability	Usability refers to the extent to which the users can easily use the software.
	reliability	Reliability refers to the abilities of the software to achieve given functions under given conditions and in given time spans.
	Efficiency	Efficiency refers to the extent to which the memory space, run time and throughput of the architecture systems has influenced on the systems.
	Security	Security refers to the abilities of the systems to keep away illegal invasion.

3.2. Develop the Trustworthiness Model

We have already built Hierarchy-tree of trusted attribute. In order for the final trustworthiness model, we need to establish the weight between attributes of each level. Using the fuzzy comprehensive evaluation methods based on AHP to solve the attribute weight, and then evaluate weight of ATNode. We refer to these Web trustworthiness attribute model called W-TM, its structure is as follows:

Primary set of attribute:
T={T_1(interface), T_2(documents), T_3(design), T_4(code), T_5(product) }

Secondary set of attribute:
T1={t_{11}(page content), t_{12}(navigation structure), t_{13}(technology environment), t_{14}(artistic style), t_{15}(uniformity) , t_{16}(humanization)},
T_2={ t_{21}(requirements specification), t_{22}(preliminary design), t_{23}(detailed design), t_{24}(development norm)}
T_3={ t_{31}(architecture design), t_{32} (module design), t_{33} (Database Design)},
T_4={ t_{41}(integrity), t_{42} (normative), t_{43} (correctness) , t_{44} (maintainability) , t_{45} (novelty) , t_{46} (reusability)},
T_5={ t_{51}(functionality), t_{52} (performance) },

Third set of attribute:
t_{12}={ t_{211}(standard), t_{212}(complete), t_{213}(reasonable), t_{214}(correct), t_{215}(clear language)}
t_{22}={ t_{221}(standard), t_{222}(complete), t_{223}(reasonable), t_{224}(correct), t_{225}(clear language)}
t23={ t_{231}(standard), t_{212}(complete), t_{233}(reasonable), t_{234}(correct), t_{235}(clear language)}
t24={ t_{241}(standard), t_{212}(complete), t_{243}(reasonable), t_{244}(correct), t_{245}(clear language)}

t31={ t_{311}(intelligibility), t_{312}(usability), t_{313}(reliability), t_{314}(safety), t_{315}(reusability) , t_{316}(maintainability) , t_{317}(expandability)}
t32={ t_{321}(integrity), t_{322}(rationality), t_{323}(reusability), t_{324}(maintainability), t_{325}(High cohesion, loose coupling) }
t33={ t_{331}(integrity), t_{332}(rationality), t_{333}(normative), t_{334}(novelty), t_{335}(reusability) , t_{336}(maintainability)}
u51={ t_{511}(integrity), t_{512}(correctness), t_{513}(consistency) }
t52={ t_{521}(ease for use), t_{522}(reliability), t_{523}(efficiency), t_{524}(safety) }

Using the fuzzy comprehensive evaluation methods based on AHP to build the attribute weight. Frist, we established the weight in the primary set. After experts discussed, for T={T1(interface), T2(documents), T3(design), T4(code), T5(product)}, their level of importance is T5> T4> T3> T1>T2, and then establish judgement matrix of a certain type, as shown in (1).

$$\begin{vmatrix} 1 & 3 & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{3} & 1 & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ 3 & 7 & 1 & 1 & \frac{1}{2} \\ 3 & 8 & 1 & 1 & \frac{1}{2} \\ 6 & 9 & 2 & 2 & 1 \end{vmatrix} \quad (1)$$

Using sum-product algorithm to find weight vector, as shown in (2).

$$W = (8\%, 5\%, 15\%, 29\%, 43\%) \quad (2)$$

Then carry on the consistency test, first calculate the CI, its formula is:

$CI = \frac{\lambda_{max} - n}{n-1}$, Where n is the number of items. CI=0.0012. And then calculate the consistency ratio CR. $CR = \frac{CI}{RI}$. Where RI called Mean Random Consistency Index, is a randomly generated number, only related to the matrix order n. when n=5, RI=1.12, CR= 0.0100<0.1. When CR<0.1, The matrix can pass consistency test, so the weight is reasonable.

Stated thus, the weight in primary set is W={8%, 3%,23%,24%,42%}(Take two significant figures). So, we can find the other level of attribute weights. Finally, the weight in all trustworthiness model is shown below:

W={8%, 3%,23%,24%,42%}
W1={30%,25%,25%,5%,5%,10%}
W2={28%,28%,28%,16%}
W21={8%,15%,32%,38%,7%}
W22={8%,15%,32%,38%,7%}
W23={8%,15%,32%,38%,7%}
W24={8%,15%,32%,38%,7%}
W3={56%,23%,21%}
W31={8%,10%,42%,28%,3%,6%,3%}
W32={28%,42%,8%,6%,16%}
W33={28%,40%,12%,5%,7%,8%}
W4={28%,10%,42%,8%,5%,7%}
W5={67%,33%}
W51={32%,58%,10%}
W52={6%,42%,38%,14%}

In this way, the trustworthiness model based on the Web application is built. This trustworthiness model can measure to trustworthiness assess for web application software. By the quantitative assessment, we can also get the appropriate level of trust.

4. Experimental Results and Analysis

By using the above trustworthiness model, we evaluate the credibility of a Web project.

4.1. Evaluation of Evidence

For attribute evaluation, the method of obtaining evidence is based on expert evaluation, tool testing, manual testing, manual inspection, *etc.*, For example, except the "human" attribute of the "interface" attribute used expert evaluation method to obtain the evaluation value, the other sub attributes are calculated by testing the Bug results.

We divided the Bug which are generated from the test into four grades（Urgent, High, Medium, Low）. Each grade of Bug has different effects on the quality of the product, so how to evaluate the Bug at different levels is particularly important. If a constant weighting factor is applied directly on the test data of different Bug levels, it is difficult to evaluate the quality of the different functions of a product. For example, the functionality of the Urgent level of Bug is not the same level as reliability in the proportion of Bug. So it is not appropriate to use the same constant factor. On the other hand, even if at the same function module of the same sub module, the number of use case is different, so the proportion of the same level of Bug should also be different. Such as 100 use cases found 10 levels for high level Bug and 1000 use cases found 100 high level Bug. Both the proportion of their defects are 1%, but we cannot simply the results of above two cases are same.

Likely, the Bug that we left over in the texting process is also an important factor affecting the quality of our testing. This is not only related to the number of Bug which we have found, but also related to over text converge, that is related to number of use cases. Well known that the number of Bug in the test has certain relationship between the Bug found in legacy software.

What we concerned is how to introduce a weighted method that is related to the number of test cases and the Bug level. Now we assume that the relation function is y=f(x) (Where y is the equivalent weighted number of defects, x is the number of defects which was directly tested). So the final weighting scheme is shown in Table 6, as follows.

Table 6. Weighting Treatment Method for Different Grade Defect

level	definition	weighted formula
Urgent	The main module of the system or the system is wrong, and there is no other temporary way to bypass the error.	$y_1 = m_1(1+\log_{10} x)$
High	The main module of the system is wrong, but there is a temporary solution can bypass the existing problems.	$y_2 = m_2(1+\frac{1}{2}\log_{10} x)$
Medium	Secondary function or document error	$y_2 = m_2$
Low	Procedures or documents are necessary to improve and perfect	$y_4 = m_4 \frac{1}{\log_{10} x+1}$

Where y_i:corresponds to the level of increase in the number of Bug.

Error! Reference source not found.$_i$: corresponds to the number of levels of Bug

x: corresponds to the total number of test cases feature items

We need to measure the formula to satisfied the above requirements but also need satisfied the size of its measurement should be in the [0,1] range. When all levels of defect data are 0, the formula corresponding to the y_i is also 0. So the final measurement formula is as (3)

$$w_j = 1 - \frac{\sum_1^4 y_i}{x + \sum(m_1 + \log_{10} x + m_2 + \frac{1}{2}\log_{10} x + m_3 + m_4 + \frac{1}{\log_{10} x+1})} \quad (3)$$

For the value of evidence for each credibility attribute index. In addition to using the expert evaluation method to obtain the evaluation value is given by the experts. The other assessment values are $100 - W_j * Bug$.

4.2. Assessment Result

According to the evaluation method of the above trusted attribute, the 49 trusted attributes of the project are evaluated, and the attribute data are obtained as Table 7.

Table 7. Project Credible Assessment Data

interface			
	page content	92	
	navigation structure	91	
	technology environment	78	
	artistic style	75	
	uniformity	93	
	humanization	85	
document			
	requirements specification		89.91
		standard	90
		complete	90
		reasonable	93
		correct	90

		clear language	75
	preliminary design		
		standard	90
		complete	85
		reasonable	90
		correct	95
		clear language	75
	detailed design		
		standard	90
		complete	92
		reasonable	90
		correct	90
		clear language	75
	development norm		
		standard	90
		complete	95
		reasonable	90
		correct	90
		clear language	75
design			
	architecture design		
		intelligibility	80
		usability	85
		reliability	90
		safety	85
		reusability	80
		maintainability	75
		expandability	85
	module design		
		integrity	100
		rationality	85
		reusability	20
		maintainability	50
		High cohesion, loose coupling	60
	Database Design		
		integrity	90
		rationality	80
		normative	80
		novelty	30
		reusability	50
		maintainability	60
code			
	integrity	90	
	normative	85	
	correctness	80	
	maintainability:8%	60	

	novelty	30	
	reusability	40	
product			
	functionality		
		integrity	100
		correctness	90
		consistency	80
	performance		
		ease for use	70
		reliability	85
		efficiency	80
		safety	85

Finally, according to W-TM trustworthiness model to calculate the amount of the credibility of the project.

T=79.1*8%+89.7448*5%+82.74588*15%+76.4*29%+88.9*43%=83.6. So the quantization credibility is 83.6. According to credible rating Table 8, this project trustworthiness level can be determined good.

Table 8. Trusted Ratings List

grade	**score**	**explanation**
excellent	90-100	Software credibility is very high
good	75-90	Good software reliability
medium	60-75	Medium in software trustworthiness
fail	Below 60	Software reliability is low

6. Conclusions

This paper established a credibility assessment model of a Web application. The results obtained by evaluating the trustworthiness of trusted attributes of projects, this evaluation result is based on software product evaluation, which provides quantitative analysis results for evaluating the trustworthiness of software projects. On the basis of this, we establish the evaluation model based on the software process and make a trustworthiness evaluation of the software project.

Acknowledgements

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

References

[1] R. Z. DU and F. ZHANG, "A trust model based on check point behaviors risk evaluation", Advanced Materials Research, vol. 403-408, **(2101)**, pp. 2102-2106.
[2] L. Shi, S. L.YANG, B. G. Yu and K. LI, "Developing an evaluation approach for software trustworthiness using combination weights and TOPSIS", Journal of Software, vol. 7, no. 3, **(2012)**, pp. 532-543.
[3] M. X. ZHU, X. X. LUO and X. H. CHEN, "A nonfunctional requirements tradeoff model", Trustworthy Software，Information Sciences, vol. 191, no. 15, **(2012)**, pp. 61-71.
[4] ISO/IEC，Information technology-security Techniques-Evaluation Criteria for IT Security. Part 1：Introduction and General Model", **(2005)**.
[5] Trusted Computing Group. TCG Architecture Overview Specification Revision 1.2, 28 April 2004[EB/OL].[2009-03-08].http：// www.trustedcomputinggroup.org.
[6] A. Algirdas, J. C. Laprie and R. Brian, "Basic concepts and taxonomy of dependable and secure computing", IEEE Trans Dependable Secure, vol. 1, no. 1, **(2004)**, pp. 11-33.
[7] H. Wang, Y. Tang and G. Yin, "The trusted mechanism of Internet software", SCIENCE CHINA：E，vol. 36, no. 10, **(2006)**, pp. 1156-1169.

[8] Y. Tang and Z. Liu, "Research progress of software credibility metrics model", Computer Engineering and Applications, vol. 46, no. 27, **(2010)**. T. Chiueh and L. Huang, "Efficient real-time index updates in text retrieval systems", New York: Stony Brook, **(1998)**.
[9] J. Zhou and M. Zhang, "Summary of software trustworthiness evaluation", Application Research of Computers, no. 10, **(2012)**, pp. 29-10.
[10] M. X. ZHU, X. X. LUO and X. H. CHEN, "A nonfunctional requirements tradeoff model", Trustworthy Software, Information Sciences, vol. 191, no. 15, **(2012)**, pp. 61-71.
[11] S. DING, X. J. MA and S. L. YANG, "A software trustworthiness evaluation model using objective weight based evidential reasoning approach", Knowledge and Information Systems, **(2011)**.
[12] A. IMMONEN and A. NISKANEN, "A tool for reliability and availability prediction", / /Proc of the 31st Euromicro Conference on Software Engineering and Advanced Applications. Washington DC: IEEE.
[13] E. AMOROSO, C. TAYLOR and J. WATSON, "A process-oriented methodology for assessing and improving software trustworthiness", / /Proc of the 2nd ACM Conference on Computer and Communications Security. New York: ACM, **(1994)**, pp. 39-50.
[14] W. J. HAN, "School Of Software EnStudy On Quality Evaluation Model Of Communication System", the 3rd International Conference on system science, engineering design and manufacturing informatization, **(2012)**.
[15] Research on quality evaluation technology of application software，The third National Conference on software testing and mobile computing, grid, intelligent Senior Forum, **(2009)**.
[16] S. Yang, S. Ding and W. Chu, "A credible software evaluation method based on utility and evidence theory", Computer Research and Development, vol. 46, no. 7, **(2009)**, pp. 1152-1159.

Authors

Wan-Jiang HAN, 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, 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.