

## **Measurement model of the software maintainability based on comentropy**

Jia Wang<sup>1</sup>, Yin-Chun Gao<sup>1</sup>, Yi-Jie Liao<sup>2</sup> and Rong Jiang<sup>1,\*</sup>

<sup>1</sup>*Yunnan University of Finance and Economics, Kunming, 650221, China*

<sup>2</sup>*Yunnan Open University, Kunming, 650223, China*

*E-mail: 416911916@qq.com*

*\*Corresponding author*

A lot of uncertain information is usually involved in the process of software system maintenance, and such information is difficult to perform quantitative analysis normally. In order to effectively solve the problem, this paper puts forward a measurement model of the software maintainability based on comentropy. A quantitative analysis algorithm about the comentropy has been introduced, then it builds up a quantitative model of predicting the software maintainability combining with qualitative analysis and calculation. In addition, the quantitative analysis algorithm model is confirmed by practical samples. The emulational result demonstrates that the method is an effective quantifiable analysis algorithm, and it can predict the maintainability of software system more accurately. Therefore this paper offers a new idea for quantifiable analysis in the prediction of maintainability of software system.

Keywords: Comentropy; Maintainability; Level Analysis; Entropy-Weight Value.

### **1. Introduction**

After the successfully development of software project, making some appropriate modifies to software are often necessary, with the process of using for users. Usually the modification to the software during the process of the operation or maintenance is called the maintenance [1]. Efficient software maintenance is very important. Software system is different from general engineering system. It is a concept system, and a logical product, so the maintenance to the software system is more difficult than general engineering system. People choose the maintainability of the software system as an important reference guidance for the software maintenance. Obviously, it's more easily to modify a higher maintainability software system than a lower. And it can reduce both the costs and risks of the maintenance, furthermore, it can improve the possibility for the successful software system maintenance.

Therefore, the measurement model which can predict the maintainability of the software system has a very vital significance to the software enterprises.

At present, there is procedural information based software maintainability metrics [2]. It defines the maintainability of the software system is key objective of all periods of software development. When maintaining software, a large number of statistical program code information is needed, great workload and time consuming. Predicting on object-oriented software maintainability based on support vector machine [3] mainly forecasts for the object-oriented software. Software maintainability evaluation model based on support vector machine [4] which is a new machine learning algorithm based on statistical learning theory, and a large number of data is needed for statistically analysis. This paper designs some statistics tables for analysis basis data information, using the important theory of the comentropy in the system science, measures the uncertainty and maintainability of the software system. The software maintainability measurement model is proposed based on comentropy. We get conclusion that this model can effectively measure the degree of software maintainability by the practice work.

## **2. The Quantitative Analysis Algorithm of Comentropy**

The information content is defined as the eliminating or reducing for the uncertainty of people knows the information in the narrowly informatics found by C. E. Shannon. American scientist John Von Neumann calls the information content which is defined by C. E. Shannon as entropy. The comentropy is the eliminating for uncertainty of information before and after communication [5].

## **3. Definition and Analysis of Maintainability**

### **3.1. Definition of system maintainability**

The maintainability of software system is the uncertain degree occurring in the process of system maintenance, and it may result in the failure of software system maintenance. And this maybe bring harm to the enterprise or even huge economic losses. Software system maintenance contains two important characteristics, they are the uncertain degree and loss degree. So this paper defines the risk maintenance as a triple  $X = (M, P, C)$ , where M means the set of the system maintenance factors, P is the probability of maintenance factors, and C is the loss degrees of system maintenance factors.

Generally speaking, the maintainability of software system is the first layer (signed as  $\beta$ ). The maintainability consists of the testability/the

intelligibility/the modifiability and the reviewability [6]. Four basic characteristics are the second layer (signed as  $\alpha$  ). Each kinds of basic characteristics connected with several maintainability indexes. Maintainability indexes are in the third layer, testability is mainly decided by the accessibility, trafficability performance, structuring and describing; And the intelligibility is mainly decided by the structuring/describing/simplicity/legitimacy/scalability and the consistency; The modifiability is mainly decided by structuring, scalability/environmental irrelevance; And the reviewability is mainly decided by the describing/simplicity reviewability of the maintenance plan /the reviewability of the journal maintenance and so on. The risk of software system maintenance will be quantitative by measuring these sub-characteristics. Fabricate the level model of maintainability, The structure of the original system as show in Fig. 1.

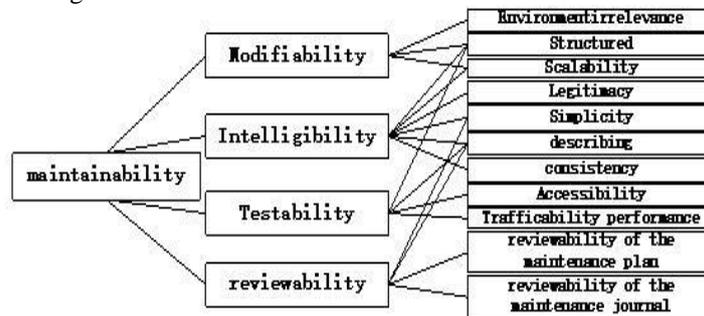


Fig. 1 The level of system maintainability

If the whole system status of the maintainability which can be maintained as the top events, the logic level figure is fabricated by analyzing the all possible reasons which causes the top events and the relationship among them. Then determines the key parts/weak link/safety requests and so on from the analyze results. Finally, evaluate the whole maintainability of the software system.

### 3.2. The comentropy values and conformity degree instructions

Based on the hierarchy graph, first of all, entropy values of the factors  $M_i$  effecting system maintenance makes the provision. The first floor of the hierarchy graph has four basic characteristic sub-items, and reflects the whole maintenance of the target system contains a number of basic characteristics of the maintenance factors, the entropy value is regarded as  $\alpha_i$  ; The second level shows that the each element of maintenance basic characteristics concludes

several specific sub-items, entropy value of the  $j$  item sub-target under the  $i$ th basic characteristics show as  $m_{i,j}$ .  $m_{i,j}$  is decided by the uncertain degree  $P_{i,j}$  and the loss degrees  $c_{i,j}$  consists in software project. Quantitative definition of the conformity degree is divided into five levels, and the specific division as shown in Tab.1 and Tab. 2.

Tab. 1 The define to maintain the uncertain degree of Conformity degree

Conformity degree	Quantitative values	Description of Conformity degree
$P_{i,j,1}$	0	the uncertain degree of system maintenance can be controlled very well, the loss is very small
$P_{i,j,2}$	0.25	the uncertain degree of system maintenance can be well controlled, less loss
$P_{i,j,3}$	0.5	the uncertain degree of system maintenance can be better controlled, loss in general
$P_{i,j,4}$	0.75	the uncertain degree of system maintenance can be controlled partly, the loss are generally big
$P_{i,j,5}$	1	the uncertain degree of system maintenance can't be controlled, the loss is the biggest

Tab. 2. The define to maintain the loss degree of compliance

Conformity degree	Quantitative values	Description of Conformity degree
$C_{i,j,1}$	0	System maintenance fund can be controlled very well, the loss is very small
$C_{i,j,2}$	0.25	System maintenance funds can be well controlled, less loss
$C_{i,j,3}$	0.5	System maintenance funds can be better controlled, loss in general
$C_{i,j,4}$	0.75	System maintenance funds be controlled partly, the losses are generally big
$C_{i,j,5}$	1	System maintenance funds can't be controlled, the loss is the biggest

In the process of actual maintenance of the software system, the uncertainty degree and damage degree are collected and settled by means of questionnaire survey methods etc. And then we ascertain the Quantitative value of Conformity degree by comparing the gaps between the uncertainty degree or the damage degree and the maintenance baseline.

#### 4. Measurement Maintainability Model Based on Comentropy

The system maintainability is divided into five grades, and they are crucial, bigger, moderate degree, smaller and negligible. The detail defines of every level as shown in Tab. 3:

Tab. 3 The define of system maintenance level

Grade	Value	Description of influence
$w_1$ crucial	0	May cause catastrophic loss, almost can consider no longer maintenance
$w_2$ bigger	0.25	May cause lots of loss, make immense efforts but useless
$w_3$ moderate	0.5	Directly affects the operation of the system, can reduce the operation efficiency of the system
$w_4$ smaller	0.75	Influence of maintenance failure is small, just smaller efforts can make up
$w_5$ negligible	1	Maintain failure miniscule, almost can not consider maintenance failure

The quantify of the specific calculation process with the overall maintainability of the software system is as follows:

The first step: obtain the conformity degree quantitative values. Suppose the entropy value of  $j$  th Subentry target under the  $i$  th basic feature is shown as  $m_{i,j}$  and the  $m_{i,j}$  is decided by the uncertain degree  $P_{i,j}$  and loss degrees  $C_{i,j}$  which software project maintenance contains.

The second step: firstly, according to formula(1), (2), (3) and (4), calculate the entropy value coefficient  $m_{i,j}$  and maintainable value  $w_i$  of the every sub-items. Calculation formulas are as follows:

$$\alpha_i = -\frac{1}{\text{Log}_2 n} \sum_{j=1}^n m_{i,j} \cdot \text{Log}_2 m_{i,j}. \quad (1)$$

$$w_i = (1 - \alpha_i) / \left( \sum_{i=1}^n (1 - \alpha_i) \right), 0 \leq w_i \leq 1, \sum_{i=1}^n w_i = 1. \quad (2)$$

The third step: similarly, the entropy value  $\beta$  of overall maintainability and the overall maintainability  $w$  can be calculated. Calculation formulas are as follows:

$$\beta = -\frac{1}{\text{Log}_2 n} \sum_{j=1}^n \alpha_i \cdot \text{Log}_2 \alpha_i. \quad (3)$$

$$w = 1 - \beta, 0 \leq w \leq 1, 0 \leq \beta \leq 1. \quad (4)$$

The fourth step: comparing the overall maintainability  $w$  of software system to the maintenance level designed by Tab. 3, the maintainability of software system can be evaluated effectively.

## 5. Analysis of model examples

### 5.1. Example calculation

Choose the concrete software system and build a hierarchy figure like Fig.1. Then calculate according to measurement maintainability model based on comentropy in section four, and analysis and evaluate the maintainability of software system. For calculating easily, the selection number of indicators is simplified in this example.

The first step: according to averaging the compliance definition from Tab. 1 and 2 and the expect grading, the uncertainty degree  $P_{i,j}$  ( 0.60,0.70,0.80,0.65,0.75,0.85,0.50,0.90,0.85,0.40,0.85,respectively ) and the loss degree  $c_{i,j}$  (0.75,0.80,0.80,0.40,0.85,0.80,0.45,0.75,0.65,0.45,0.90, respectively ) of the bottom eleven sub-item indexes can be obtained .

The second step: both the entropy value coefficient  $m_{i,j}$  and overall maintainability  $w_i$  of every sub-item objective in the second layer can be calculated according to the formula Eq. (1-4).

The third step: both the entropy value  $\beta$  and overall maintainability  $w$  of the overall maintainability of software system can be calculated, and the values are  $\beta=0.8115$  and  $w=1-\beta=0.1885$ , respectively.

In addition, to illustrate that, the initial compliance quantitative values uses two precision. However, four precision is needed for emulational calculation. So, the error and error accumulation in the calculation process can effectively be avoided and reduced.

### 5.2. Analysis of maintenance risk

For the analyzing the overall maintainability of the software system, comparing the definition of maintenance risk level, the actual overall successful maintenance risk of software systems can be got  $w=0.1885$ , and the maintenance is  $0 < w < 0.25$ , and near to 0.25, Within the small possibility of successful maintenance, so the maintenance success risk is high.

The overall maintenance risk of comprehensibility at the second layer is 0.1378, which belongs to a very critical level, and it is the major factor of the overall software maintenance. However, the modifiability, testability, and reviewability are all about 0.3, the risk of overall system maintenance is more

average and the relative intelligibility is more smaller. The testability of the overall maintainability is 0.3112, and the value is bigger, the influence for overall maintenance is smaller.

The simulation results of actual data show that, the conclusion get by analyzing comentropy of the maintainability agree with the actual situation, and indicate that the measure algorithm is reasonable.

## 6. Conclusion

The maintainability factors of software systems are analyzed, then a quantitative calculation method of software system maintainability based on comentropy is suggested, and also a complete analysis model is formed building on the basis of hierarchical graph. The initial data of the model is real and reliable, the maintenance risk of software system is analyzed use comentropy model, the level of maintainability of software system is assessed in advance, furthermore, it can response and management to the maintainability more scientific.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant Nos. 61263022 and 61303234), Science and Technology Innovation Talents Program of Yunnan Province(Grant No. 2015HB038).

## References

1. Renjie Zheng. Software Engineering(Advanced) [M].Beijing: Tsinghua University Press,1999:105-107(In Chinese).
2. Xiaoya Sun, Jing Chen. Procedural Information Based Software Maintainability Metrics [J]. Shandong Science,2010,23(4):68-71(In Chinese).
3. Lijin Wang, Xinxin Hu. Predicting on Object-Oriented Software Maintainability Based on Support Vector Machine [J].Journal of Beihua University,2010, 11(3):282-285(In Chinese).
4. Xuejuan Chen, Meisen Pan, Chaoyang Lei. Software maintainability evaluation model based on support vector machine[J]. Computer Engineering and Design,2008,29(3):566-570(In Chinese).
5. Dongchuan Sun, Fuyong Lin. Systems Engineering Introduction [M].Beijing: Tsinghua University Press,2004,97-98.
6. Yang Wang, Zongmin Zhao, Haitao Wu. Metrics on Software Maintainability Improvement via Refactoring [J]. Microcomputer Applications,2009,30(10):36-42(InChinese).