Research on Risk Assessment of Software Testing Project Based on Fuzzy Comprehensive Evaluation Method

Qian Yang

Logistics Engineering, Xidian University, Xi'an 710126, China.

1434648814@qq.com

Abstract

As an important means to ensure the quality of software, software testing has attracted more and more attention.There are risks in software testing projects, and risk management is an indispensable part of the software testing process.If we can prioritize risk assessment and develop a positive response plan for possible risks, we will minimize the risk or reduce the risk caused by the risk when the risk comes.Based on the risk identification of software testing, this paper establishes the risk evaluation index system of software testing project by using the theory of fuzzy comprehensive evaluation method.In the risk assessment process for software testing, quantify and determine the level of risk. Finally, the effectiveness and feasibility of the method are illustrated by an example.

Keywords

Software Testing, Risk Assessment, Fuzzy Comprehensive Evaluation.

1. Introduction

An objective and fair evaluation of the software can be obtained by testing the software for comprehensive comprehensive functions and performance. Software testing has been widely used in many fields, such as in the field of e-commerce, aviation, and so on. As an independent engineering activity, software testing has certain special laws and specific technical characteristics, and there are also inevitable risks[1]. For example, the constant change in software requirements has led to an extension of the development cycle, resulting in insufficient test time, over-expenditure of test costs, and unsatisfactory test quality. Therefore, the management and control of software test project risks become more and more important[2]. Among them, the quality of risk assessment has a direct impact on the management and control of risk. In this paper, the risk of software testing project is researched and evaluated effectively, and the fuzzy comprehensive evaluation method is used to quantify the risk impact indicators of software testing projects, then the objective score is given[3].

2. Evaluation Index System and Fuzzy Comprehensive Evaluation

2.1 Evaluation index system

The risk of software testing refers to the potential difficulties or problems that may arise during the software testing process. The occurrence of software testing risks may result in insufficient testing of software products or inaccurate or unsatisfactory test results [4].The risk of software testing is inevitable, it exists in the entire software testing process, so the management of software testing risks is very important.

Risk identification is the basis of risk assessment. The purpose of risk identification is to determine which risks may affect the project.[5]Risk identification should be carried out early in the project, and test-related risk identification should be carried out at the beginning of the test plan to clarify the factors that pose a threat to the project, and to facilitate the development of plans and strategies to avoid risks and reduce risks. For software test projects, the test process generally includes test requirements and planning, test design and implementation, test execution and test summary, and the risk of the test project runs through the entire test process[6].Through the identification of risks in the various processes of the software testing project, risk classification is carried out, and an evaluation index system is established. The software test project risks are divided into four primary indicators such as technical risk (U1), environmental risk (U2), progress risk (U3), and management risk (U4), and then divided into 12 secondary indicators under the first-level indicators. The evaluation index system is shown in Table 1.

Table 1 Risk factors in the evaluation index system

Primary indicator	Secondary indicators
Technical Risk (U1)	Test Requirements Analysis (U11)
	Technical Method Master (U12)
	Test Case Design (U13)
Environmental Risk (U2)	Space Environment Difference (U21)
	Hardware Environment Difference (U22)
	Software Environment Difference (U23)
Progress Risk (U3)	Progress Estimate (U31)
	Problem Recognition and Modification (U32)
	Test Demand Change (U33)
Management Risk (U4)	Test Resource Guarantee (U41)
	Personnel Management (U42)
	Test Process Management (U43)

2.2 Fuzzy comprehensive evaluation

Once the risk factors have been identified, the risks need to be assessed. In software testing projects, it is difficult to directly and accurately determine the probability of occurrence of various risk factors and the severity after occurrence through risk quantification. In order to accurately describe the intensity and extent of the impact of each risk factor, this study introduces a fuzzy comprehensive assessment theory for analysis[7]. The basic idea of fuzzy comprehensive evaluation is to use the principle of fuzzy linear transformation and the principle of maximum membership degree to consider various factors related to the things being evaluated. Therefore, a reasonable comprehensive evaluation is given, which reduces the subjective influence to a certain extent, making the evaluation more scientific and reasonable[8].

The process of fuzzy comprehensive evaluation is: First, consider the evaluation target as a fuzzy set (factor set U) composed of multiple factors. Secondly, set the level of review that these factors can be selected to form a fuzzy set of reviews (assessment set V). Then, the degree of attribution of each single factor to each review level (fuzzy matrix) is obtained. Finally, according to the weight distribution of each factor in the evaluation target, the quantitative solution value of the evaluation is obtained by calculation (fuzzy matrix synthesis). Proceed as follows:

(1) Determine the fuzzy comprehensive evaluation factor set U and the evaluation set V: Determine the factor set U (establish a system of evaluation indicators) to determine which aspects of the evaluation, U = {u1, u2, ... un}. The judgment set V is a set of comment components, V = {v1, v2, ... vm}.

(2)Single factor evaluation: Judging the single factor ui(i=1,...,n) to obtain the fuzzy set on V (ri1, ri2,...,rim). So it is a fuzzy mapping from U to V:

$$f: U \rightarrow F(V) \quad u_i \mapsto (r_{i1}, r_{i2}, \ldots, r_{im}) \tag{1}$$

The fuzzy map f determines a fuzzy relationship $R \in u_{n \times m}$, called the evaluation matrix, which is composed of all F sets evaluated for single factors.

(3) Determine the weight coefficient of each factor in the concentration factor: Since the status of each factor may not be equal, it is necessary to weight each factor. The weight distribution of each factor is represented by the F set A=(a1, a2, ..., an) on U. The weights should satisfy both normal and non-negative, ie ∑aj=1 and aj≥0. The methods for determining the weight coefficient are mainly Analytic Hierarchy Process (AHP), Delphi, Prior Knowledge, and Least Squares.

(4) Calculate the judgment result and make an evaluation judgment: Through the synthesis of the single factor evaluation result and the weight vector, the final evaluation result is obtained, B=A·R, B=(b1, b2,..., bm). The final result of the fuzzy comprehensive evaluation is a numerical vector. The evaluation and analysis of this numerical vector can be used to obtain qualitative or quantitative conclusions. The methods of evaluation judgment and analysis generally have the maximum membership degree method, the fuzzy distribution method, and the central average defuzzification method.

3. Case Analysis

The fuzzy comprehensive evaluation method is used to evaluate the risk degree of a software test project, and the risk score is obtained to determine the risk level. The specific application steps of the fuzzy comprehensive evaluation method are as follows.

3.1 Establish a set of judgment factors and a set of judgments

For the risk assessment of the software test project, a set of evaluation factors is established, as shown in Table 1. For the evaluation set, we are divided into 4 levels, namely V={high, higher, medium, low}.

3.2 Single factor evaluation

For the single factor evaluation, the project uses the Delphi method, and several experts form an evaluation group to evaluate each single factor against the risk indicator system factor set and risk severity evaluation set established in the previous article. And the evaluation results are counted. We have established a fuzzy relationship from the risk factor set U to the risk comment set V, and obtained the membership degree of each single factor in the risk indicator to the evaluation set, as shown in Table 2.

Table 2 Membership vector table

Factor set	Main factor	R1			R2			R3			R4		
	Sub-factor	R11	R12	R13	R21	R22	R23	R31	R32	R33	R41	R42	R43
Single factor evaluation	High	0.382	0.212	0.118	0.126	0.657	0.461	0.339	0.261	0.512	0.173	0.314	0.260
	Higher	0.345	0.380	0.453	0.397	0.254	0.410	0.321	0.387	0.306	0.127	0.307	0.411
	Medium	0.203	0.303	0.320	0.208	0.046	0.106	0.208	0.231	0.153	0.254	0.251	0.037
	Low	0.081	0.116	0.128	0.280	0.054	0.034	0.143	0.132	0.040	0.457	0.139	0.014

3.3 Determining the weight set

This paper chooses AHP to determine the weight of indicators at all levels. According to the principle of analytic hierarchy process, the same level indicators are compared in pairs according to the 1~9 scale method to construct the judgment matrix[9]. The maximum eigenvalue of the judgment matrix and the corresponding eigenvector are calculated, and the obtained eigenvectors are normalized to obtain the weight vector A=(a1, a2, ..., an) of the index. Using the same method, you can get the weight vector of each level. In order to judge whether the obtained weight coefficient is reasonable, according to the AHP principle, it is also necessary to perform consistency check on the judgment matrix. The random consistency ratio CR=CI/RI is calculated according to the deviation consistency index: CI=(lmax-n)/(n-1) (where lmax is the maximum eigenvalue of the judgment matrix and n is the order of the judgment matrix) and the random consistency index:RI. When CR≤0.1, the judgment matrix can be considered to have satisfactory consistency, otherwise the judgment matrix needs to be adjusted and has satisfactory consistency.

According to the above steps, the judgment matrix is established layer by layer, and the weights of the risk indicators are as follows: The weight of the project-level indicators is A=(0.18, 0.49, 0.37, 0.16). The technical risk weight of the secondary indicators is A1=(0.42, 0.27, 0.35). Test environment risk weight A2 = (0.25, 0.63, 0.14). Schedule risk weight A3 = (0.21, 0.56, 0.25). Management risk weight A4 = (0.46, 0.32, 0.23).

3.4 Calculate risk fuzzy evaluation results

(1) First-level fuzzy comprehensive evaluation: Using the formula B=A·R, the membership degree vector of each criterion is obtained.

$$B1 = A1 \cdot R1 = \begin{bmatrix} 0.42 & 0.27 & 0.35 \end{bmatrix} \times \begin{bmatrix} 0.382 & 0.345 & 0.203 & 0.081 \\ 0.212 & 0.380 & 0.303 & 0.116 \\ 0.118 & 0.453 & 0.320 & 0.128 \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} 0.242 & 0.386 & 0.273 & 0.108 \end{bmatrix}$$

The same can be calculated: $B2 = A2 \cdot R2 = \begin{bmatrix} 0.497 & 0.322 & 0.209 & 0.112 \end{bmatrix}$,

$B3 = A3 \cdot R3 = \begin{bmatrix} 0.332 & 0.354 & 0.190 & 0.249 \end{bmatrix}$,

$B4 = A4 \cdot R4 = \begin{bmatrix} 0.238 & 0.245 & 0.093 & 0.106 \end{bmatrix}$.

Therefore, the matrix of membership metrics is obtained:

$$R = \begin{bmatrix} B1 \\ B2 \\ B3 \\ B4 \end{bmatrix} = \begin{bmatrix} 0.242 & 0.386 & 0.273 & 0.108 \\ 0.497 & 0.322 & 0.209 & 0.112 \\ 0.332 & 0.354 & 0.190 & 0.249 \\ 0.238 & 0.245 & 0.093 & 0.106 \end{bmatrix} \tag{3}$$

(2) Second-level fuzzy comprehensive evaluation: Find the membership degree vector of the total target.

$$B = A \cdot R = \begin{bmatrix} 0.18 & 0.49 & 0.37 & 0.16 \end{bmatrix} \times \begin{bmatrix} 0.242 & 0.386 & 0.273 & 0.108 \\ 0.497 & 0.322 & 0.209 & 0.112 \\ 0.332 & 0.354 & 0.190 & 0.249 \\ 0.238 & 0.245 & 0.093 & 0.106 \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} 0.410 & 0.369 & 0.188 & 0.137 \end{bmatrix}$$

Normalize B to get $\begin{bmatrix} 0.375 & 0.334 & 0.173 & 0.126 \end{bmatrix}$.

3.5 Evaluation results

The maximum membership degree method and fuzzy distribution method can be used to obtain qualitative results for the fuzzy comprehensive evaluation result vector. The central average defuzzification method can be used to give quantitative conclusions[10].This paper chooses the central average defuzzification method to analyze the judgment results. The expression of the central average defuzzification method is:

$$\sum_{l=1}^{n} y^l w_l \Big/ \sum_{l=1}^{n} w_l \tag{5}$$

In equation (5), yl represents the center of the fuzzy set of each linguistic variable, and wl represents the degree of membership of each linguistic variable.

Using the central average defuzzification method, it is necessary to determine the central value of each linguistic variable that is representative of the risk level represented by the evaluation set. By using the Duffel method to solicit a number of expert opinions, with a maximum risk of 100 points, we determined that the central value of the risk level is {low (30), medium (50), higher (70), high (90)}. When V<90, it is judged as high probability risk. When 70<V<90, it is judged as high risk. When 50<V<70, it is judged as medium probability risk. When V<50, it is judged as low probability risk.

The central average defuzzification method is used to calculate the final risk quantification:

$$\begin{bmatrix}0.375 & 0.334 & 0.173 & 0.126\end{bmatrix} \times \begin{bmatrix}90 & 70 & 50 & 30\end{bmatrix}^T = 69.56 \tag{6}$$

The score is in the range of $50 < V < 70$, so it is judged as a general medium probability risk.

4. Conclusion

Based on the establishment of the third-party software testing risk assessment index system, this paper uses the fuzzy comprehensive evaluation method to evaluate the risk of third-party software testing projects. The model is validated by examples to determine the risk score and determine its risk level. As the evaluation results give quantitative values, determine the degree of impact of the risk on the software test project, and provide a basis for determining the risk management security strategy of the software test project and selecting an effective defense plan.

References

[1] Lina Zheng, Wei Wang, Yue Zhou. Analysis of the Development Status of Third-party Software Testing in China[J]. Software Industry and Engineering, 2012,17(5):38-41.

[2] Weitao Li, Jian Zhang, Yixiao Wang. Research on Software Project Risk Assessment Based on Grey System Theory[J]. Computer Technology and Development,2013,23(5):124-126.

[3] Hall P. Knowing the Odds[J]. The Software Testing and Quality Engineering Magazine, 2003, 5 (2) .

[4] Wiegers K E. Know Your Enemy: Software Risk Management[J]. Software Development Magazine, 1998.

[5] Bingyuan Cao. Applied Fuzzy Mathematics and System[M]. Beijing: Science Press, 2005.

[6] Jie Zhao. Management System Engineering[M]. Beijing: Science Press, 2006.

[7] Jun Yang, Sufen Zhang. Software Test Project Risk Assessment Method Based on Fuzzy Evaluation[J]. Research and Design,2015,41(5):14-17.

[8] KOLLMANN T, KUCKERTZ A. Evaluation Uncertainty of Venture Capitalists'investment Criteria[J]. Journal of Business Research, 2010 (7) .

[9] Srihari J, Jinson J E, Juning K. Fuzzy Analytic Hierarchy Process Based Group Decision Support System to Select and Evaluate New Manufacturing Technologies[J]. The International Journal of Advanced Manufacturing Technology, 2007, 32 (11) :1253-1262.

[10] BORGONOVO E, PECCATI L. Uncertainty and Global Sensitivity Analysis in the Evaluation of Investment Projects[J]. Journal of Production Economics, 2006 (1) .