

Research on Evaluation of Component-Based Software Testing

Xueyun Ji and Zihui Che

Department of Mathematics and Computer, Baoding University, Baoding 071000, Hebei, China

Abstract: The research object of our work is evaluating program for the component-based software testing. As an important part into the software engineering, software testing is the primary instrumentality to guarantee the quality and reliability of the software. With the Component-Based Software Development (CBSD) method becoming a popular method of software engineering, people pay more and more attention to the quality of component software. Therefore, the component-based software testing is an important method to guarantee the quality of component software. There are many factors existing in the component-based software testing process, in order to avoid some errors and defects and improve the testing process, we need evaluate the validity of component-based software testing. Based on the discussion of the quality characteristics of the software component, the study studies the workflow and the relevant measures which apply to evaluate the validity of component-based software testing.

Keywords: Component, component-based software testing, validity of software testing, quality, workflow

INTRODUCTION

Software quality (Chen, 2008) is the consistent level that between the software product and the needs which defined explicitly and implicitly. Software quality evaluation national standard ISO 14958 defines software quality as “the combination of software properties, the ability that software can meet the users’ demands which specified explicitly or potential”. Software testing is the assurance of software quality. Software testing (Hou and Cui, 2008) can not only find out errors that existing in the software, but also can help finding the defects during the software development process, through analyzing the causes and stages that make errors, to improve the software development process and the software quality.

With component technology become mature and the CBSD (Hu and OuYang, 2008) method uses widely, software developers enjoy the advantages which bring by component technology, such as development rapidly, higher reusability and so on. And at the same time, they pay more attention to the quality of component software and the problem of component software testing. There are many studies about component software testing now, researchers have proposed many feasible methods and testing strategy, such as the testing method based on component metadata (Ma *et al.*, 2006), the testing method based on UML (Unified Modeling Language, UML) (Shang and Zhang, 2006), the testing method based on interface mutation (Andres and Macario, 2009) and so on. Software testing occupy a large proportion in the process of software development, the percentage of workload of testing in the total workload is usually

between thirty and forty percent. In order to improve the testing process and use the developing cost effectively, it is necessary to evaluate the validity of software testing. At present, in the field of studies about the evaluation of software testing, there are many research results, for example, the method of evaluation of software testing quality combined with fuzzy theory (Yao *et al.*, 2007), the dynamic evaluation method based on multi dimensional test coverage (An *et al.*, 2010) and an evaluation model of validity of software testing based on Bayesian theorem (Wang, 2007).

The study studies the component-base software testing and studies the evaluation of the validity of component software testing to ensure the quality of component software.

THE QUALITY OF SOFTWARE COMPONENT

Component is the smallest unit in the component software system; its quality influences the quality of component software directly. The industry pays more and more attention to the importance of the quality of software component. For software developers, they concern about evaluating the quality of software component to improve the quality of software component product; for the person who reuse software component, they are more worried about how to evaluate the quality of software component to select suitable software component from a mass of component library.

Software component has the quality characteristics which are also suitable for the generally software.

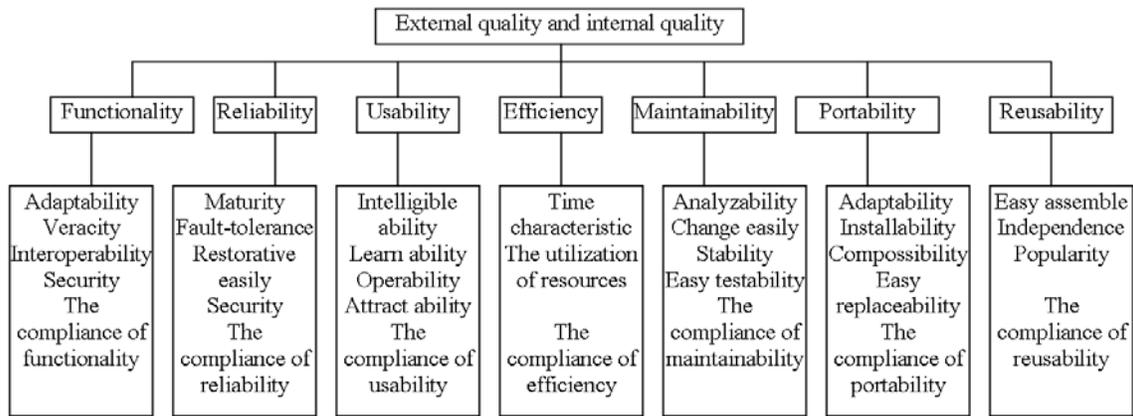


Fig. 1: The model of external quality and internal quality of software component

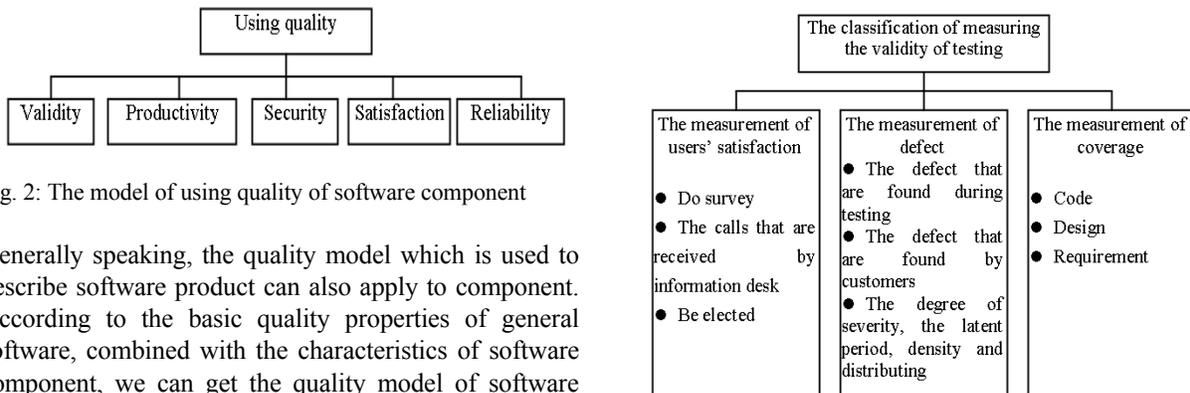


Fig. 2: The model of using quality of software component

Generally speaking, the quality model which is used to describe software product can also apply to component. According to the basic quality properties of general software, combined with the characteristics of software component, we can get the quality model of software component. Similar to the general software, the quality model of software component also divides three parts; they are external quality model, internal quality model and using quality model. We get the external and internal quality model through the following process: on the basis of the basic quality properties in the software, combining with the component characteristics, we add a quality property which is related to component, which is reusability, as shown in Fig. 1.

Using quality is the quality from users' view, its acquiring is dependent on obtaining the required external quality and the external quality is dependent on obtaining the required internal quality. Form users' view, if the component can be reuse or not and the times of reusing reflect the trust degree of users use the component directly. The using quality model of software component is as shown in Fig. 2. We can see from Fig. 2, on the basis of the original using quality properties of software, it increases a using quality property, which is the reliability of software component. The reliability has two factors: reusing degree and choosing degree. Reusing degree identifies the reusing reliability of software component and choosing degree identifies the expected selection of software component, it shows a proportion that potential user will use this component.

Fig. 3: The classification of measuring the validity of testing

THE MEASUREMENT OF VALIDITY OF TESTING

The development of component and component software bring along the study and development of component software testing. The main purpose of evaluating the validity of testing is to evaluate testers' work and use the evaluation results to improve testing process. During component software testing process, it usually has some invalid aspects; the goals of evaluation include identifying invalid problems in order to take repair measures. The classification of common means which measure the validity of testing is as shown in Fig. 3.

The measurement of customers' satisfaction is a method that measures afterwards, this method can't get the measurement results before the product is in sales, installation and using.

The common measurement method of validity of testing is build around defect analysis. Defect analysis is the following series of activities (Xin, 2009). It does necessary collection with defects which are generated in the process of software development and running (Agrestl and Evanco, 1992). Then classify the

information about the defects, do summary statistics, calculate the analytic indicator and compile the analytical reports. Defect analysis provides some indexes of software reliability (Aroul and Lavergne, 1996). The indexes provide the judging standards to reveal the trend or the distribution of defects. The common defect analysis indexes include the detection rate of defect, the density of defect, the elimination rate of defect and the loss of defect and so on:

$$D\rho = \frac{Dn}{Fn} \tag{1}$$

We define (1) is the computing formula for the density of defect, which Dp denotes the density of defect, Dn denotes the number of defects and Fn denotes the number of code lines or function points.

$$DRE = \frac{Dm}{Dm + Ndn} \tag{2}$$

We define (2) is the computing formula for the elimination rate of defect, which DRE expresses the elimination rate of defect, Dm expresses the number of defect that are found during testing process and Ndn expresses the number of defect that are undiscovered.

$$DL = \frac{Dn \times Fg}{Tdn} \tag{3}$$

We define (3) is the computing formula for the loss of defect, which DL means the loss of defect, Fg means the weight values of preclinical stages that find out defect, Tdn means the total number of defects.

The measurement of coverage is a practical measurement method of the validity of testing. The coverage can measure the testing sets or the integrity of actual executing testing.

THE WORKFLOW AND THE RELVANT MEASURES OF EVALUATING COMPONENT SOFTWARE TESTING

The workflow of evaluating the validity of component software testing: During the process of evaluating the validity of testing, we need to focus on two aspects: the aim of evaluation and the required information categories that can achieve the aim of evaluation. Making right aim can guide the whole evaluating process and the collection of accurate information can ensure the evaluating process smoothly. Figure 4 shows the workflow of evaluating the validity of testing.

The evaluating work of validity is begin when the testing process is finished, its input information are the ending executing results of testing stage, these input information include the following types: the number of executed testing, the resources that are consumed in testing, the tool that are used in testing, the defect that are discovered, the size of software that is tested, the number of days that is used to repairing defect, the defect that are not repaired, the defect are found in operation that are undiscovered in testing, the phases that find out defect and the name of discovered defect and so on. The executing process of evaluating the validity contains seven aspects: that are making the aim of evaluation, determining the contents of evaluation, making the responsibility of measurement, choosing the method of evaluation, determining the required facts, collecting the data of evaluation and evaluating the validity of testing. After the whole executing process, according to the quality checklists, we can inspect the evaluating process, part quality inspection items are as shown in Table 1. Then the reasonable evaluation can output the improved testing process, otherwise it need to return to the start of the executing process, remake the aim of evaluating and execute the evaluating process again.

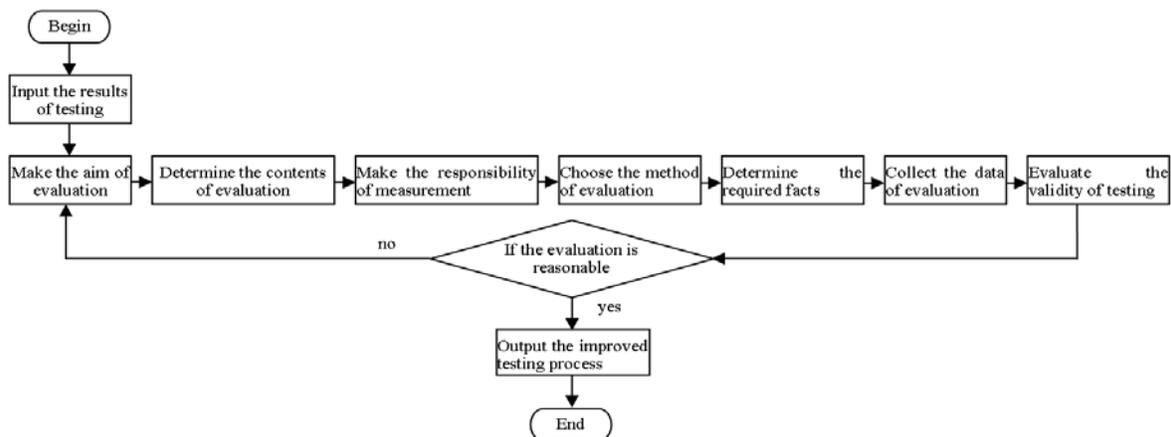


Fig. 4: The evaluating flow of the validity of software testing

Table 1: A part of checking item of quality

Serial number	The description of project
Project 1	Whether the management department support the continuous improving the process of testing or not
Project 2	If the resources that used to improve the process of testing are available
Project 3	If the results of testing accumulate with time growing
Project 4	If the result of testing include the various of component items which are described in input parts
Project 5	If the analysis result is reasonable
Project 6	If the executing analysis is based on the characteristic of component software testing
Project 7	If the analysis result is used in improving the testing process

Table 2: The measures to evaluate the testing project of component software

Serial number	Measures	Using measures	The description of measures
1	The time cost in testing	The degree of testing	The time cost during testing process
2	The personnel involved in testing	The degree of testing	The number of people that participate in testing
3	The changing rate of the schedule of testing	The degree of testing	The deviation rate of the actual testing schedule compared to the schedule in planning
4	The number of testing	The degree of testing	The number of testing and the scale of the tested system
5	The component that is tested	The degree of testing	The number of tested component and the total number of component
6	The cost of testing process	The resources that consumed during testing	The cost of testing and the total cost of the whole system
7	The cost that locate defect	The resources that consumed during testing	The cost of testing and the number of defect that is located during testing process
8	The budget that achieved	The resources that consumed during testing	The expected cost of testing and the actual cost of testing

Table 3: The measures to evaluate the testing process of component software

Serial number	Measures	Using Measures	The description of measures
1	The errors that are found in product	The validity of testing	The number of errors that are found in product and the scale of the system
2	The defect that are found during testing	The validity of testing	The defect that is located during testing and the total defect of the whole system
3	The defect of interface that are found during testing	The validity of testing	The defect of the interface of component that are assigned by testing
4	The validity of business by testing	The validity of testing	The loss cost by problem and the total resources of the system
5	The times that the component reusing	Involving system	The times of the component was reused
6	The validity of the unit testing of component	The validity of testing	The cost of unit testing and the number of errors are found during this process
7	The validity of the integration testing of component	The validity of testing	The cost of integration testing and the number of errors are found during this process
8	The Validity Of The Testing Stage	The Validity Of Testing	The cost of testing and the number of errors are found during debugging

The measures of evaluating component software testing: We need a measuring standard to do objective evaluation for the validity of testing program, by analyzing the measures, the evaluators can adopt targeted measures, record the results of evaluation, so it is necessary to establish indexes system to evaluating component software testing. The traditional software testing programs have three aspects of measurement: the measurement of project, the measurement of process and the measurement of product. Combining with the traditional measures, this study establishes the measures to evaluate the testing of component software system. Table 2 and 3 show a part of measures.

CONCLUSION

From component software testing, the study discusses external quality model, internal quality model and using quality model of software component, on the research of the measurement of validity of testing, it proposes the workflow to evaluate the validity of component-based software testing and at the time it gives the related measuring indexes. Using the workflow

which we proposes to evaluate the validity of component software testing, it can improve the process of component-based software testing, play the role of component-based software testing bitterly and improve the quality of component-based software.

REFERENCES

- Agrestl, W.W. and W.M. Evanco, 1992. Projecting software defects from analyzing software designs. *IEEE T. Softw. Eng.*, 18: 988-997.
- An, J., G. Wang, S. Liu and J. Zhu, 2010. Dynamic evaluation method based multi-dimensional test coverage for software testing. *J. Softw.*, 21(9): 2135-2147.
- Andres, F. and P. Macario, 2009. Testing-based process for evaluating component replace ability. *Electr. Notes Theor. Comput. Sci.*, 236(2): 101-115.
- Aroul, M.E. and C. Lavergne, 1996. Generalized linear models in software reliability parametric & semi-parametric approaches. *IEEE T. Rel.*, 45(3): 463-470.
- Chen, P., 2008. Evaluation of software quality. *Softw. Guide*, 7(1): 32-34.

- Hou, H. and D. Cui, 2008. Research on assurance of software quality based on software testing technology. *Sci. Technol. Inform.*, 13: 58.
- Hu, G. and S. OuYang, 2008. Component-based software product development and investigation. *Control Eng. China*, 15(1): 75-81.
- Ma, L., Y. Lu and M. Liu, 2006. Application research of metadata on generating component-based software test case. *Mini-Micro Syst.*, 27(12): 2228-2231.
- Shang, X. and Y. Zhang, 2006. Research of UML-based generating test case for component integration testing. *Comput. Eng.*, 32(1): 96-98.
- Wang, D., 2007. An evaluation model of validity of software testing based on Bayesian theorem. *Mod. Bus. Trade Ind.*, 19(11): 268-269.
- Xin, F., 2009. The reason and the analysis for the bug appeared in software. *J. Hebei Acad. Sci.*, 2009(26): 28-30.
- Yao, Y., X. Liu and S. Huang, 2007. An assessment method for fuzzy partial ordering relation-based software testing quality. *J. Univ. Electr. Sci. Tech. China*, 36(3): 503-509.