

Innovation and Practice of the Course of Software Testing

Y.F. CHEN & J. WANG & Y.Q. PENG

School of Software, Guangdong Ocean University, China

L.X. PENG*

** School of Computer Science, Guangzhou University, China*

ABSTRACT: To the actuality of teaching on software testing, we introduce some innovations: modeling course architecture, developing some unit experiments and a comprehensive experiment, inducing CDIO (Conceive, Design, Implement and Operate) idea of engineering education, actively carrying out cooperation with enterprises, constructing the course website. The practices show that, combining the theory with practice of software testing, these reform measures can effectually improve the students' enthusiasm. For the same time, the students increase their innovation ability from studying and training in the advance engineering idea

KEYWORD: software testing; teaching; CDIO (Conceive, Design, Implement and Operate)

1 INSTRUCTIONS

With the development of information technology, the talents requirement of software testing is rapidly increased in recent years. However, the graduates hardly get their appropriate jobs. Thus training practical talents of software testing is one of the most important tasks that we must reform to the specialty of software engineering.

For the software becomes more and more complex and larger in scale, the technologies of software testing are developing as well as the software development methodologies. In paper [1], a method called "Nature-inspired techniques for conformance testing of object-oriented software" was presented. A modeling framework was developed in paper [2] for software reliability testing. The book [3] discussed the art of software testing. The paper [4] presents ongoing research at the University of Hertfordshire on the use of computer-adaptive tests (CATs) in Higher Education. The paper [5] introduces the automatic testing environment for multi-core embedded software.

2 STATUS ANALYSIS OF TEACHING

In our university, Guangdong Ocean University, there are some matters as following in teaching of software testing course:

1. Most of students are lower interested in studying software testing. Software testing is an investigation conducted to provide stakeholders with information about the quality of the product

or service under test [6]. To them, the software development is a constructive work, however software testing is destructive, so there is less attraction in studying software testing for students.

2. The tutorials of software testing pay more attention to theoretic but not practice, thus the students only learn the surface of theoretic and do not know how to utility. Bannon presents an in-depth survey of "developing software as a human, social and organizational activity" [7]. So we should make students reflective thinking about the approaches of software testing and taking more practice.

In order to solve the above problems, we first guide the students to take interest in courses. Through the network, we search a lot of recruitment information and show them to students. Then students realize the large amount of demand for software testing talents, and take great interest in the course. At the same time, we strengthen the study and analysis of case teaching, and strengthen the practice of training, so that students can make practice better with theory. In our class, we inspire students discussing, exchanging their thinking and clearing the fuzzy concepts.

3 INNOVATION OF TEACHING

3.1 Modeling course architecture, strengthen foundation of theoretic

The first task of software testing is to understand the basic concepts and to grasp the correlation

technologies, tools and methods. We construct the modeling course architecture, which core is technologies of software testing. It contains 6 teaching modules: basic concepts, methodology, black-box testing, white-box testing, and mobilization testing and object-oriented testing, together with 21 teaching units. These modules can be neatly assembled as the real situation and the content can be continually updated according to the technology development of software testing. The teachers should lead the students to think, instead of one-way instill the knowledge into adults.

In our course, we must make cases analysis in order to strengthen the understanding of theories. For example, when teaching “the equivalence class method” of black-box test, we adopt examples of “NextDate()” and “determine the shape of a triangle” to show students how to construct the equivalence classes. In the choice of cases, we choose those popular cases to study, such as case of functional testing of mobile phone.

3.2 Developing some unit experiments and a comprehensive experiment

The practical teaching of software teaching is extension of theoretical course and an important part of training complex software engineering talents. The paper-a survey of software testing practices [8] shows that the importance of testing-related training is increasing. We developed unit experiments and comprehensive experiments. That progressive approach enables students to master the application of theoretical knowledge. The unit experiments include static testing, unit testing and white-box testing. Each unit experiment is always simple, which aims at checking whether the students master the special testing approach. The comprehensive experiment is a software product based phone information management system. The students make different kinds of test, such as unit testing, functional testing and performance testing on it.

The comprehensive experiment makes the students utilize the global software testing techniques and integrated approaches. It is useful to improve the talent of applying testing techniques in project. The students are group into team(3~5 adults) to complete the comprehensive experiment. From making the comprehensive experiment, the students realize the principle that a test case’s function is to elicit information quickly and efficiently. The students use different strategies, either static or dynamic testing, either white-box testing or black-box testing. Moreover the students make use of the automatic testing tools. The students also made some discusses on the problems: What are the tidy and effective test case? And then they can finally make more effective test with fewer test cases. The team

pattern can also exercise the capacity of management and cooperation.

The procedure of making the comprehensive experiment is shown as Figure 1.

Step1. Introduce project. The testing products are given. And necessary documents including software requirement specification (SRS), user manual etc., are also given. And testing requirement can be provided in advance or made according to experiences of testers.

Step2. Plan the arrangement of test. Make a simple of schedule of time, personal and material. The important thing is to make testing strategies and ending standard of testing in this phase.

Step3. Design testing cases. As a tester, the most difficult work is to choose effective testing methods to find more software bugs. The tester will use white-box testing method for codes with complex logic. When make a white-box testing, a code coverage tool can help to cover all code branches easily. To ensure sufficient functionality testing, black-box testing is also made, such as Equivalence partitioning, boundary value analysis. These techniques are frequently combined to design more sufficient test cases without any knowledge of internal implementation.

Step4. Execute testing cases. Testers execute all the testing cases, record any bugs found and report them to the developers. When the bugs are fixed, the testers have to make regression testing to ensure that modifications will not bring new bugs.

Step5. Analyze testing results. After testing all testing cases, testers write final reports, objective estimate the quality of product. The final reports always include bugs trend, bug distributions of type and severity.

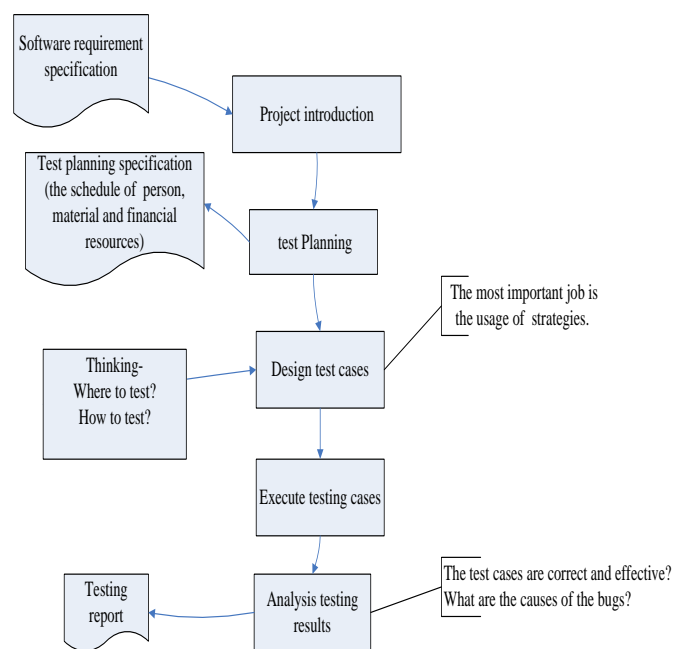


Figure 1. Procedure of making comprehensive experiment

3.3 *Introduce the education concept of CDIO engineering and establish of training architecture integrated with experiment and practice*

The education concept of CDIO engineering is integrated with Conceive, Design, Implement and Operate, which has been widely used in international training engineering talents at present. The advanced international CDIO is introduced into the teaching of comprehensive practice, which combines practice teaching and application closely. In the stage of conceiving, we try to help the students analyze the requirements of testing and establish the ideal of "the requirements of testing are come from the users and the experience of testers". For there aren't actual users in teaching, we can achieve the requirements of testing from "simulating the requirements of users and the actual working environment" and from the experience of testers. For example, when making performance testing on a web-based project, the common querying time is 5s as the experiences of testers. When estimating the number of concurrent users, we usually set that the value as 20-50% of the users. Through the training of CDIO, students improved their analysis abilities of testing requirement and designing effective test cases using different test strategies for different requirements.

3.4 *Actively carrying out cooperation with enterprises, to meet with the talents of software testing*

As the provincial model software college, its aim is culturing practical software developers. So, actively carrying out cooperation with enterprises can make students experience the technology demands of software testing from the enterprise. Then the students learn how to utilize theoretical knowledge into actual practice. For owning rich theoretical knowledge but absent with experiences of developing software project in practice, the instructors are poor at guiding students to practice. On the other hand, cooperative enterprises have mature corporate culture, the rich resources of the project, the staff professionalism training experience and proven operating experience. For the project manager and technical director have wealth of experience in software project development, school-enterprise cooperation achieves the schools and businesses "complementary advantages, resource sharing, mutual benefit and common development" win-win results. By showing project workflow and details implementation of large-scale software testing, cooperative enterprises enable students to learn the practical experience of the test and to increase the management experience of software testing project.

3.5 *Constructing course website and developing the ability of independent learning*

The networked teaching is a modern mode of higher education, such as "web-based teaching in software engineering" [9]. Constructing the course website and uploading course exercises, courseware, and test questions, which allow students to freely choose the time to learn. Limited knowledge taught in the classroom is far from satisfying the needs of different levels of students, students can greatly expand the breadth and depth of knowledge of software testing, by learning a wealth of information on the Web site. The website also provides to the link the famous websites such as the software test network (51testing.com) website, China Software Testing Alliance (<http://www.51sqa.com/>) etc. These linked sites provide a lot of testing the exchange of experience, the latest test theory, research and various test tools in motion. They can fully mobilize the enthusiasm and initiative of the students as platforms that software testers constantly learn and progress.

4 CONCLUSIONS

Software testing course is a both application and practical courses, which should be both strengthened in teaching. From teaching and practical experiences, we make a number of tentative teaching reform measures: modeling course architecture, developing unit experiments and comprehensive experiments, inducing CDIO idea of engineering education and constructing the course website. The curriculum reforms have set a number of training, which make students cultivate rigorous thinking habit and be easy in designing efficient test cases. We have made four years practice at our university; the results show that these reform measures can effectually improve the students' enthusiasm, practical ability and innovation ability.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61100150, Educational reform Project of Guangdong Ocean University under Grant No. XJG201309.

REFERENCES

- [1] A. Bouchachia, R. Mittermeir, P. Sielecky, S. Stafiej and M. Ziemiński, "Nature-inspired techniques for conformance testing of object-oriented software", *Applied Soft Computing*, pp.730, 2010.
- [2] Y.C. Chang, W.L. Huang, "Software release policies on a shot-noise process model", *Applied Mathematics and Computation*, pp.746, 2005.

- [3] Myers, G.J., "The Art of Software Testing", John Wiley & Sons Inc., New York, 2004.
- [4] M. Lilley, T. Barker, C. Britton, Cem Kaner, "The development and evaluation of a software prototype for computer-adaptive testing", Computers & Education, pp.109, 2004.
- [5] Chorng-Shiuh Koonga and Chihhsiong Shihb et al., "Automatic testing environment for multi-core embedded software—ATEMES", Systems and Software, vol.85, pp.43, 2012.
- [6] "Exploratory Testing", Quality Assurance Institute Worldwide Annual Software Testing Conference, 2006.
- [7] Bannon L., "Developing software as a human, social and organizational activity. 13th Annual Workshop of the Psychology of Programming Interest Group", Bournemouth, 2001.
- [8] Garousi V., Zhi J., "A Survey of Software Testing Practices in Canada", The Journal of Systems and Software, 2010.
- [9] N. Papaspyrou, S. Retalis, S. Efremidis, G. Barlas, E. Skordalakis, "Web-based teaching in software engineering", Advances in Engineering Software, vol. 30, pp. 901, 1999.