# Research On Teaching Methods And Tools Of Software Testing

1st Honghong Chen
*School of Computer and Software Engineering*
*Xihua University*
Chengdu, China
chenhh@mail.xhu.edu.cn

2nd Xu Wang
*ChengDu College*
*University of Electronic Science And Technology of China*
Chengdu, China
34106831@qq.com

3rd Liangguang Pan
*School of Computer Science and Engineering*
*University of Electronic Science And Technology of China*
Chengdu, China
1032102576@qq.com

*Abstract*—**With the rapid development of software industry, software testing, as an important means of software quality assurance, has been paid more and more attention by software enterprises. It has become an urgent need to train professional software testing talents, so software testing teaching is of great significance. This paper analyzes the current situation and existing problems of software testing curriculum teaching, discusses and improves the teaching methods of software testing courses, and develops a simple suspicious codes recognition tool based on abstract grammar tree and state machine for the difficulties of teaching content which based on defected modes testing , so as to help improve the quality of classroom teaching and better train professional software testing talents suitable for market demand.**

*Keywords—software testing, curriculum teaching, defected modes, suspicious codes, recognition tool*

## I. THE SIGNIFICANCE OF SOFTWARE TESTING TEACHING REFORM

With the development of computer technology and social demand, software plays an increasingly important role in modern society. At the same time, due to the increasing complexity and scale of the software, there are some serious software quality problems. Software testing, as an important means to ensure software quality and improve software reliability, has been paid more and more attention by software enterprises[1].

Mastering software testing knowledge is a basic quality of software industry practitioners, and it is also the basic requirement for colleges and universities to train software professionals. China's software testing industry is in its infancy, the shortage of basic talents in software testing, to some extent restricts the development of the software industry[2]. At the same time, due to the enterprise's requirements for software quality more and more stringent, the gap in software testing talent is still increasing. To this end, the training of software testing talents has become a hot spot in education and training in the software industry[3]. At present, many colleges and universities have opened software testing courses, but due to the limitations of the traditional curriculum teaching model, there is still a certain gap between the cultivation of talents and the actual needs of enterprises. Therefore, the software testing course should also be reformed to adapt to the development and needs of the software testing industry market.

## II. THE CURRENT SITUATION AND PROBLEMS OF SOFTWARE TESTING CURRICULUM TEACHING

### A. Teacher-centered

The curriculum teaching form of Chinese colleges and universities is mainly to classroom explanation, after-school practice, this way has the characteristics of high efficiency and high targeting[4], can transfer professional knowledge to a large number of students in a short period of time, and strengthen understanding through after-school targeted practice. But this way is a teacher-centered mode and only the teacher to students unilateral knowledge transfer, students in the classroom more time is receiving knowledge, it is difficult to timely in-depth understanding of knowledge and feedback ideas. For students, it's just inheritance of knowledge not exploration of knowledge. According to the experimental study of teaching effect, the teaching effect of classroom teaching is poor, and only 5% of the content learned after 24 hours[5].

### B. Insufficient students' ability to learn on their own

In the process of teachers unilaterally imparting knowledge to students, the requirement of students' independent learning is gradually weakened. The low student participation is difficult to stimulate students to think positively about the content of the course, which is not conducive to the students' digestion of knowledge, and students are prone to lose their interest in learning[6]. At the same time, coupled with the students' own willpower and other reasons, leading to the problem of lack of independent learning ability. As we all know, software testing may appear in the actual application of some new problems, and the lack of self-learning ability will lead to the relatively easy to solve known problems, and new problems can not be solved in a timely manner.

### C. Lacking of practicality in curriculum teaching

In the actual teaching process, there may be too much emphasis on the teaching of test theory methods, but neglect the cultivation of practical ability[7]. Most students do not have practical software project development experience, lack of engineering concepts, so it's abstract and difficult for them to understand the content of the course [8]. Most students can't test a software product after completing the main content of the course, thus not achieving the goal of integration. Because of the weakness of practical teaching, it is difficult for students to apply flexibly to the understanding of the theoretical basis of the software test curriculum.

Therefore, it is difficult to develop the hands-on and teamwork required by the enterprise.

## III. A STUDY ON THE REFORM OF SOFTWARE TEST TEACHING METHODS

### A. Students lectures and thematic discussions

In the course of teachers' traditional teaching, there is a problem of one-way transmission of knowledge[5], and students lack in-depth thinking, which will weaken their ability of independent learning. In order to avoid continuous knowledge-irrigation teaching, resulting in the students' learning burnout, it is possible to use the combination of teacher teaching, student practice and topic discussion. According to the characteristics of the content of the software test curriculum, the teaching content is divided into teacher teaching and students' practice, the difficulty of software test theory is taught by the teacher, and the practical and open content of software test is mainly practiced and discussed by the students in the classroom. The students' participation in the classroom is increased by using the teaching method of "flip classroom" with students as the main body[3]. Finally, by the teacher to summarize, ask questions, supplement, answer questions, the whole classroom teachers and students to discuss together.

Because students will carry out a lot of self-study preparation, such as consulting materials, summarizing, making PPT, etc. before carrying out the course and discussion. Therefore, through this teaching model, on the one hand, can strengthen students' ability to learn independently and understand the curriculum knowledge more deeply. On the other hand, it also improves the student's document organization ability and oral expression ability. The teaching effect is good.

### B. Reinforcing curriculum practice

In the practical teaching link, based on the real project, with the test post requirements as the core, choose the appropriate software test method and tool for a real software project, complete the development of the test plan, test case design and test execution, fill in the software defect record report and write the test report. Usually the form of team organization, division of labor completed, as a team to carry out assessment, through the implementation of the project so that students to deepen the understanding of the software testing process, and initially have the use of knowledge to carry out practical problems of the test process organization and test case design ability[8], exercise and improve student communication, Communication and expression skills, as well as teamwork skills, to develop students' sense of responsibility and professional ethics, for the training of software testing personnel to lay a more solid foundation.

### C. Developing and using assisted teaching tools

With the progress of science and technology, teaching aids are more and more abundant, research shows that the use of aids for teaching, than the traditional theoretical knowledge explanation more vivid, students are more easily accepted and understood.

In order to enhance the students' understanding of part of the content of the defected modes testing, this paper takes the teaching content of the defected modes of the suspicious codes as an example, and develops a simple tool for the identification of the suspicious codes based on the abstract syntax tree and the state machine. Through the teaching assistant tool, input the codes to detect defects, and display various types of defects, so that students can understand various codes defects more vividly.

The following is to take Java codes as an example, the principle and implementation of simple defect pattern detection tools. First, the source codes are analyzed, then the abstract syntax tree is generated, the custom defect pattern is loaded, and finally the defects are detected throughout the abstract syntax tree nodes and the results are marked. The overall framework is shown in Figure 1.
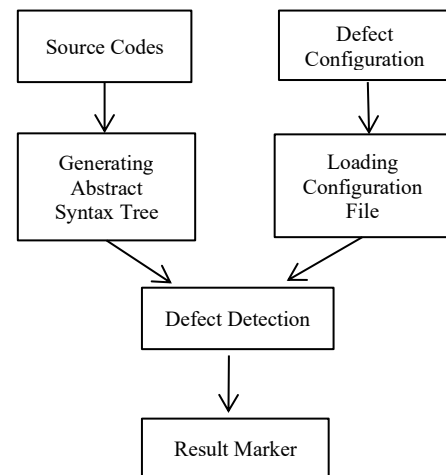


Fig.1. Overall Framework

- Building an abstract syntax tree

An abstract syntax tree is an abstract syntax structure of a tree that represents source codes, and each node in the tree represents a structure of the source codes. Building an abstract syntax tree is of great significance to codes analysis, as shown in Figure 2.
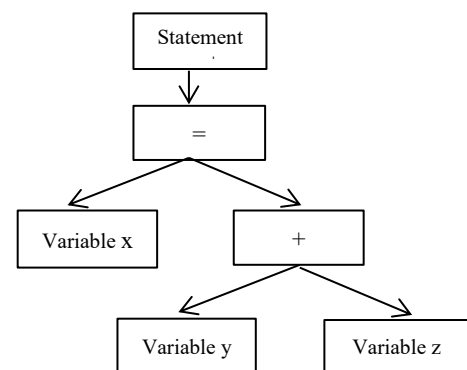


Fig.2. Abstract Syntax Tree Schematic

The tool uses Eclipse AST to generate an abstract syntax tree, and the syntax structure in the Java source program is represented as an ASTNode node. The built abstract syntax tree includes Statement Block, Expression Statement, While Statement, If Statement, Empty Statement, Nodes such as Assignment.

- Defining defected modes

Taking into account scalability, the tool provides the flexibility to add defect detection rules to more defect pattern detection by implementing the

BaseChecker interface and defining configurations. Several common suspicious codes defect patterns are defined to meet some basic functions[9], as shown in Table 1.

TABLE 1 DEFINITION OF SOME DEFECTED MODES

| Defect Name | Defect Definition | Testing Rules |
|---|---|---|
| Self assigned defect | The value variable of the assignment statement is the same as the assigned variable, such as a=a; | Detect symupts for the consistency of the Expression nodes of the Assignment node, if consistent lying to the self-assignment defect. |
| Hollow block defect | The block of codes are empty, such as {} | Detect the number of Statement nodes in the Block node, if empty indicates an empty block defect。 |
| Empty statement defect | Residual semicolon after statement, such as a=1;; | Detect the presence of the Empty Statement node if there is a flaw in the prompt empty statement. |
| Equivalent branch defect | The codes are the same on different paths, such as conditional statements, such as if(a>1){ b=1; }else{ b=1; } | Detect whether the then Statement node of the If Statement node and the else Statement node are consistent, if consistently indicate the equivalent branch defect. |
| Suspect assignment defect | The conditional judgment part assigns a value to the Boolean variable, such as if(a=1){ b=1; } | Detect the parent node type of the Assignment node if the If Statement node or The While Statement node indicate a suspected assignment defect。 |

- Detecting and marking defected modes

Eclipse AST uses the ASTVisitor visitor class to traverse the abstract syntax tree, while the ASTVisitor class provides different overloaded visit() methods depending on the node type. After overloading the visit() method by inheriting the ASTVisitor class and overloading the visit () method based on the defined defect detection rule, traverse the abstract syntax tree. Custom defect patterns are called to detect and store the detection results during the traversal of the node.

At the end of the test, then according to the stored test results the source codes are marked and displayed. As shown in Figure 3 and Figure 4 (yellow label as a defect)



Fig. 3. Source codes to be detected



Fig. 4. Test Results (Yellow is Defective)

## IV. CONCLUSION

This paper discusses and studies the reform of software testing teaching methods, adjusts the teaching form of curriculum according to the characteristics and current situation of software testing classroom, develops auxiliary teaching tools, enriches vivid classroom teaching content, attaches importance to the cultivation of students' practical ability, and enhances students' software testing ability from different ways in order to better meet the needs of enterprise testing talents. In the future, with the constant change of software testing technology, the teaching methods of software testing curriculum also needs to continue to reform and improve, keeping pace with the market demand.

## REFERENCES

[1] Bing Fu. Software quality and testing [M]. Beijing: Tsinghua University Press, 2017

[2] Shaoming Zhu. "Research on the problem driven teaching mode of software testing course" [J]. China University Teaching, vol.338(10), pp.32-36, October 2018

[3] Li Hua,Qiong Gu,Xia Huang,Bin Ning. "The Blended Teaching Mode of Software Testing Course in the Internet Plus Context" [J]. Journal of Neijiang Normal University, vol.33 (02), pp.95-99, February 2018

[4] Xiaolin Ju,Xiang Chen,Wangzhi Wen,Yan Mei. "Construction of software testing curriculum teaching system under the background of Internet plus" [J]. Computer Education, vol. 05, pp.112-116, May 2017

[5] Li Yin, Jin Dai, Linhui Liu. "Reflection and Exploration on the TeachingMode of Software Testing Course". [J]. Science Tribune, vol.35(12), pp.113-114,December 2019

[6] Zhiguo Ding. "On the teaching reform of software testing course under the background of MOOC" [J]. Computer Education, vol. 03, pp.46-48, March 2016

[7] Shiliang Zhang. "Exploring the teaching reform of Software Testing" [J]. Journal of Ningde Normal University(Natural Science), vol. 25(04), pp.415-418, April 2013

[8] Min Wang,Yaguang Chen. "The exploration of software test experiment teaching" [J]. Computer Education, vol. 03, pp.53-56, March 2011

[9] Wei Liu. "Source codes based suspicious codes and bad programming practice detection" [D]. Beijing University of Posts and Telecommunications. 2009