

Experimental Teaching Design of Software Testing Process Management

Zhiyan Zhang

School of Software Engineering, Anyang Normal University, Anyang 455000, China

Abstract

The experimental teaching of software testing course attaches great importance to technical practice. This paper mainly discusses the teaching content design of software testing process management experiment. Through this experimental teaching content, students can grasp all aspects of software testing in general, further understand the software testing activities to be completed in all aspects, and connect the scattered knowledge points of the course to form an organic whole, which is conducive to the further digestion and absorption of the course knowledge.

Keywords

Software Testing Course; Test Process Management; Experimental Teaching Reform.

1. Introduction

Software testing course is a professional core course for software engineering majors in colleges and universities, which includes both theoretical teaching and experimental teaching. For a long time, the experimental teaching of software testing course has placed great emphasis on technical practice, mainly including unit testing, integration testing, functional testing, security testing, performance testing and other aspects. In fact, with the continuous promotion of higher education reform and the gradual promotion of engineering education certification, all-round training of students' abilities must be paid attention to. On July 15, 2022, the certification standards for engineering education issued by the China Association for Engineering Education Professional Certification put forward clear graduation requirements in three aspects: "project management", "individuals and groups" and "communication". Therefore, if the experimental teaching of software testing course only includes technical practice, it can no longer meet the training requirements of the new era, so it is necessary to add new experimental teaching content to promote the achievement of new curriculum objectives.

As a new experimental teaching content, the software testing process management experiment can enable students to grasp all aspects of software testing as a whole, further understand the software testing activities to be completed in all aspects, and connect the scattered knowledge points of the course to form an organic whole, which is conducive to the further digestion and absorption of the course knowledge. At present, there are few experiments on software testing process management, so relevant experimental teaching is designed. Through the implementation of this experiment, the new graduation requirements in the engineering education certification standards are supported.

2. Experimental Design

With the help of relevant test management tools, the experiment content can realize the management of major test activities such as test plan development, test requirement analysis, test case design, defect handling and tracking, and test report preparation. As shown in Figure 1. Because students can master the software testing process in general only through

experiments, the relevant content should be simplified compared with the actual software testing process in enterprises. For example, to develop a test plan, you only need to reflect the key points of the test plan in the prepared documents, and test requirements analysis and test case design only need to be done for a small function point of the system under test.

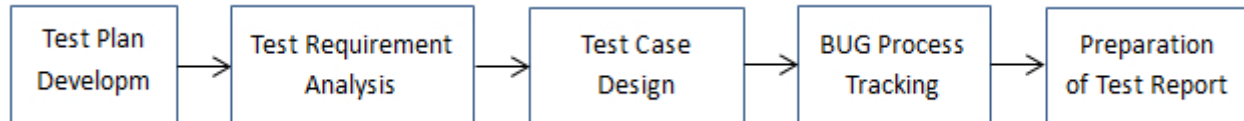


Fig 1. Main test activities

In order to cultivate students' team awareness and communication ability, the experiment was conducted in groups. In groups of 3 to 5 people, the role of each student is determined through discussion within the group. The relevant roles mainly include project manager, developer, test team leader, test case designer, and test executor. In the case of a small group, one person can play multiple roles.

There are many software test management tools, including open-source tools such as TestLink, Bugzilla TestRunner and Mantis, and commercial tools such as HP ALM, JIRA and IBM Rational Test Manager. Many software test management tools also have great differences in functions. Some functions are relatively simple, such as only limited to the management of test cases and defects, and some functions are relatively powerful, supporting complete software test process management. Recently, a domestic test management software, ZenTao, is widely used in China. As shown in Figure 2. This software has comprehensive functions and supports complete software development project management, including strong support for various management activities in software testing.

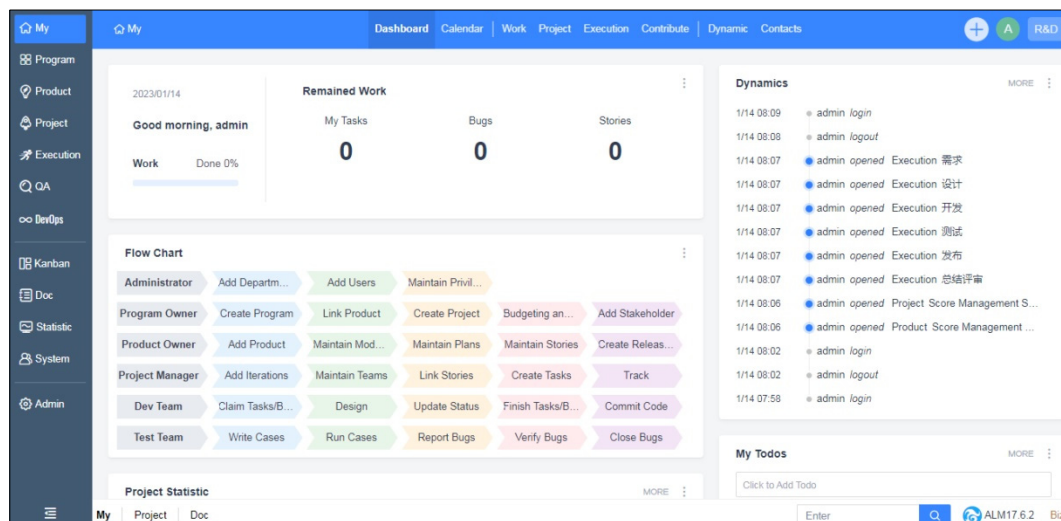


Fig 2. Software interface of Zentao

3. Experimental Process

3.1. Test Plan Development

The test plan can be jointly formulated by the team members, and then input into the system by the test team leader. Test plan documents mainly include test objectives, test scope, test strategy, test resources, test schedule, test risks, etc. The test objective is to make a high summary of the purpose of the test. The scope of the test needs to be clear about which aspects of the test are carried out and to what extent. The test strategy is to solve the problem of how to do it. Test strategy belongs to technical definition. Define in the test strategy how to achieve

the test objectives in the test plan and the relevant technologies that may be applied. Test resources need to describe the software and hardware resources and human resources used in the test process. The test schedule needs to determine the start and end dates of each test stage. The test risk shall describe the test risk and solutions. As shown in Figure 3.

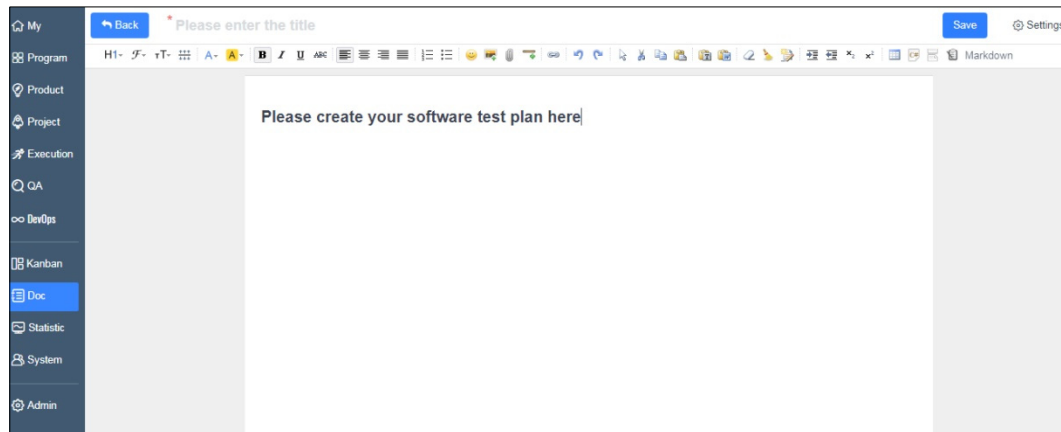


Fig 3. Test plan development

3.2. Test Requirement Analysis

Test requirements and software requirements are two different concepts. Test requirements refer to what is tested and to what extent. Software test requirements are derived from software requirements. The acquisition of software test requirements can be collected from the software requirements specification. Software test requirements can be divided into functional test requirements and non-functional test requirements. In the course of the experiment, we mainly practiced the functional test requirements. As shown in Figure 4.

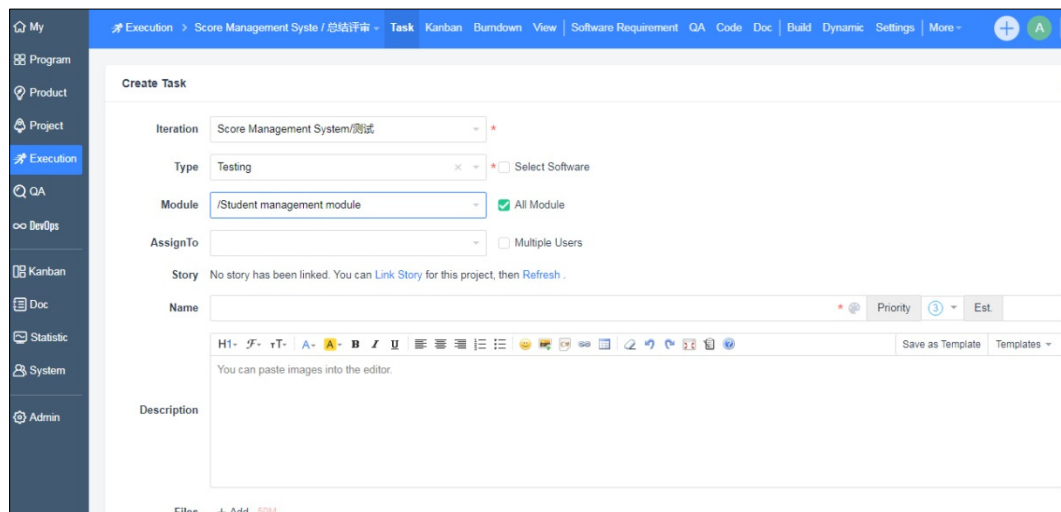


Fig 4. Test requirement analysis

3.3. Test Case Design

After extracting test requirements, students in charge of test case design can start test case design activities according to the developed test strategy. For functional testing, the methods commonly used for test case design include equivalence class, boundary value, decision table, causality diagram, orthogonal experiment, state transition, scenario design, etc. Students can choose one or two methods to practice. The test case should include the test title, use case attributes, importance level, preset conditions, operation steps, expected results and other main contents. As shown in Figure 5.

The screenshot shows the 'Add Case' form in the QA module. The form includes the following fields and sections:

- Product:** Score Management System
- Module:** /student management module
- Type:** Feature
- Linked Story:** (empty)
- Title:** login test case
- Prerequisite:** (empty)
- Steps Table:**

ID	Step	Expectation	Action
1	enter user name	☐ Group	+ - ✕
2	enter password	☐ Group	+ - ✕
3	click login	☐ Group	+ - ✕
4		☐ Group	+ - ✕
- Tags:** (empty)

Fig 5. Test case design

3.4. BUG Process Tracking

During the execution of test cases, as shown in Figure 6. Students are mainly asked to practice the tracking management of software defects found, including filling in the defect report by the students in charge of the test role, viewing the defect report by the students in charge of the development role and modifying the defect status. As shown in Figure 7.

The screenshot shows the 'login test case' execution form. It includes the following fields and sections:

- Prerequisite:** (empty)
- Table:**

ID	Step	Expectation	Result	Details
1	enter user name		Pass	
2	enter password		Pass	
3	click login		Pass	
- Save:** (button)

Fig 6. Execution of test cases

The screenshot shows the 'BUG process' form. It includes the following fields and sections:

- Product:** Score Management System
- Module:** /
- Project:** (empty)
- AssignTo:** A:admin
- Discovered by:** (empty)
- Type:** CodeError
- Title:** login test case
- Steps Table:**

ID	Step	Expectation	Action
1	enter user name		
2	enter password		
3	click login		
4			
- Prerequisite:** (empty)
- Tags:** (empty)

Fig 7. BUG process

3.5. Test Report Preparation

Based on the data collected during the test and the analysis of the final test results, the test report forms a document containing the test process and test results. The test report is the final document output of the test stage. A detailed test report should contain sufficient defect analysis information, and also include the evaluation of product quality and test process. As

with the test plan, students can only reflect the main points when they practice writing the test report.

4. Conclusion

Software testing is a core course for software engineering majors. Reasonable arrangement of experimental content can help students quickly improve their software testing ability and meet the needs of social development. Carrying out software testing process management experiment can improve students' overall understanding of software testing work, connect the fragmented knowledge points of the course, and facilitate the digestion and absorption of the course content.

References

- [1] Ron Patton. Software Testing. China Machine Press, 2017.
- [2] Lixin Liang, Haisheng Li, et al. Software Testing Technology and Project Case Tutorial. Tsinghua University Press, 2022.
- [3] Wenjie Du, Wei Wang. Software Test Tutorial. Tsinghua University Press, 2022.
- [4] Jing Gao, Li Zhang, et al. Software Testing and Quality Assurance. Tsinghua University Press, 2022.
- [5] Information on: <https://www.csdn.net>.
- [6] Information on: <https://www.zentao.net>.