< Online Teaching System>

SQA Test Plan (STP)

Version: < 1.1 >

< 09/01/2021 >

Revision History

Date	Version	Description	Prepared / Revised By
<30/12/2020>	<1.0>	First version of the SQA	潘凯航，张溢弛，康大凯，李楠，聂俊哲，张琦
<09/01/2021>	<1.1>	Newest and Final Version	张溢弛

Table of Contents

1 Introduction 5

1.1 Document Terminology and Acronyms 5

1.2 References 5

2 Target Test Items 6

2.1 Test Inclusions 6

2.2 Test Exclusions 8

3 Test Approaches 9

3.1 Test Identification and Justification 9

3.2 Conducting Tests 9

3.2.1 Build Acceptance Testing 9

3.2.2 Functionality Testing 9

3.2.3 Regression Testing 10

3.2.4 Volume/Performance/Failover testing 10

3.3 Test Automation Strategy 11

3.4 Defect Management 11

3.5 Test Metrics 11

3.6 Reporting 13

4 Entry and Exit Criteria 13

4.1 Test Execution Entry Criteria 13

4.2 Test Execution Exit Criteria 13

4.3 Suspension and Resumption Criteria 14

5. Environmental Needs 14

5.1 System Hardware 14

5.2 Software Elements in the Test Environment 15

6. Responsibilities, Staffing, and Training Needs 15

6.1 People and Roles 15

6.2 Staffing and Training Needs 17

7. Key Project/ Phase Milestones 17

8. Risks, Dependencies, Assumptions, and Constraints 18

8.1 Risk 18

8.2 Dependency 19

8.3 Assumption 19

8.4 Constraint 20

1 Introduction

1.1 Document Terminology and Acronyms

SQA -- Software Quality Assurance

BVT -- Build Verification Testing

Black Box Testing

Functionality Testing

Regression Testing

Final Integrated Build Testing

1.2 References

[1]*Software Engineering: A Practitioner's Approach 8th Edition*, Roger S.Pressman, McGraw-Hill Education

[2]*Software Requirements 3rd Edition*, Karl Wiegers, Joy Beatty， Microsoft Press

[3]*ISO 9000 for Software*

[4] [G22] "Online Teaching System" Project Proposal

[5] [G22] "Online Teaching System" Feasibility Analysis Report

[6] [G22] "Online Teaching System" Project Charter

[7] [G22] Quality Assurance Plan for the Online Teaching System

[8] [G22] Requirements Engineering Plan for the Online Teaching System

[9] [G22] "Online Learning System System" Vision and Scope Document

[10] [G22] Requirements Specification for "Online Teaching System

[11] [G22] System Design Plan for the Online Teaching and Learning System [12] [G22] [G22] System Design Plan for the Online Teaching and Learning System

[12] [G22] System Coding and Advance Planning for the Online Teaching System

[13] [G22] Requirements change control meeting protocol for "Online Teaching System

[14] [G22] "Online Teaching System" Requirement Change Plan

[15] [G22] "Online Teaching System" Training Plan

2 Target Test Items

2.1 Test Inclusions

- Common Page
  - Index Page
  - Open Class Page
  - Help Page
- Module of User Info
  - Register
  - Login
  - Change password
  - View user info
  - Edit user info
  - Manage authority
- Module of Course Info

  - User views all courses
  - User searches courses
  - User views detail of courses
  - User enters a course
  - Admin syncs course info between educational administration system and online teaching system
- Module of Notice
  - User views released notices
  - Teacher & TA release notices
- Module of course homework
  - Student views homework
  - Student submits homework
  - Teacher assigns homework
  - Teacher edits homework
  - Teacher marks homework
  - Teacher handles appeal
  - Teacher divides class into groups
  - Teacher appends group member
  - Teacher views group info
- Module of course materials
  - Student downloads materials

  - Teacher uploads materials
  - Teacher updates materials
  - Teacher deletes materials
- Module of course discussion
  - User posts
  - User deletes posts
  - User comments
  - User replies

2.2 Test Exclusions

- Module of test and exam
  - Teacher assigns tests
  - Student answers tests
  - System marks tests
- Module of question library
  - Teacher appends questions
  - Teacher edits questions
  - Teacher deletes questions
- Module of online experiment
  - Student conducts online experiments

3 Test Approaches

3.1 Test Identification and Justification

- Due to the low priority of module of question library and module of online experiment, we may choose not to implement them considering the time limit is tight.
- Module of test is similar with module of course homework, and is under development, thus we don't test it now.
- Other modules like common page, module of user info, module of course info, module of notice, module of course homework, module of course materials module of course discussion are complete in development and will be tested.

3.2 Conducting Tests

3.2.1 Build Acceptance Testing

- To implement build acceptance testing(BAT), we will check functions according to user manual and review all possible defects in the report library. What's more, we'll do setup, configure and test for the RELEASE1.0 version of our online teaching system.
- We'll set up a set of test cases to ensure the application programs stable and can be tested completely. If building acceptance testing fails, developers will fix bugs and release new versions.

3.2.2 Functionality Testing

- The purpose of functionality testing is to check if the system has correctly covered all functions, and if the functions correspond to the requirement documents. Testers will use strategies like equivalence class partition and boundary value analysis to design use cases and then test each module.

Functionality testing typically involves following steps:

1. Identify functions that the software is expected to perform.

2. Create input data based on function's specifications.

3. Determine the output based on function's specifications.

4. Execute the test case.

5. Compare the actual outputs with expected outputs.

- In functionality testing, we list all the main functions of the house management system, and design the data and operation set of the input and output in the test according to the test case specification. Then we will execute these test cases and compare the actual output with our Expected to be consistent. If it does not match, we will correct the code and re-test.

3.2.3 Regression Testing

- Regression Testing is to confirm that a new release of software version doesn't adversely affected existing features. It's done to make sure that new changes should have no side effects on existing functionalities. And it ensures that the features still works.

- After each bug is identified and then fixed, we must perform regression testing to ensure our changes do not affect the rest of the system.

3.2.4 Volume/Performance/Failover testing

- Volume testing is to analyze the system performance by increasing the volume of data in the database. With the help of Volume testing, response time and system behavior can be studied when exposed to a large volume of data.

- Volume testing is necessary for our online teaching system. We must test the large amount of users, high concurrency, and huge amount of data in advance to allow the system to operate normally under such circumstances.

3.3 Test Automation Strategy

- Test Automation is to make use of tools to manage and control tests as well as analyze results of test. It can help the team to conduct a large amount of test cases. It's the future of software testing to replace human by test automation tools. For some modules inside the system, like personal information, material downloading, discussion in course, we can simulate a user and his/her operation using test automation tools.

Here're some test automation tools our group uses:

- JTEST
- BugZilla
- Jmeter
- Rational Test Manager

3.4 Defect Management

We'll make use of open-sourced detect tracing systems to record defects in our system and export to a test report after testing.

We'll continually trace and fix the conditions of defects during development.

At the same time, defect management helps further regression testing.

3.5 Test Metrics

We'll select some basic test metrics to measure the development quality and test quality of our work. They're as follows:

- count of test cases
- count of passed test cases

- count of un-passed test cases
- count of bugs
- acceptability of bugs
- count accepted bugs
- count of fixed bugs
- count of bugs can be fixed
- count of critical bugs
- expected date
- actual date

Moreover, we'll make use of the data to calculate some derivate metrics:

- accept rate=count of passed test cases/count of test cases
- fail rate=1-accept rate
- fix rate of bugs=count of fixed bugs/count of bugs
- accept rate of bugs=count of accepted bugs/count of bugs
- can-be-fixed rate of bugs=count of bugs can be fixed/count of bugs
- validness=count of bugs found/count of bugs
- cover rate of test=count of test cases run/count of test cases
- cover rate of requirements=count of requirements covered/count of requirements
- cover rate of requirements(weighted)=sum of weight of requirements covered/count of requirements
- time difference=expected date-actual date

3.6 Reporting

We'll generate the following documents:

- [G22] test plan
- [G22] test report

And we'll submit test case tables and reports about bugs found.

4 Entry and Exit Criteria

4.1 Test Execution Entry Criteria

- Developers have done most coding work and have conducted basic unit test in development environment
- Online Teaching System can bypass build acceptance testing and smoke testing
- Requirements in SRS can be fundamentally covered
- Test plan and test cases have passed the review

4.2 Test Execution Exit Criteria

- The project being tested has reached basic requirements of the SRS
- All test cases have been conducted
- All bugs found have been recorded, and critical bugs have been fixed
- All bugs have been developed a solution
- Functional testing and Non-functional testing have been conducted
- The final test report has been generated

4.3 Suspension and Resumption Criteria

- If a critical defect or many small defects affect the testing, the testing should be suspended
- If the project funding is tight, the testing should be suspended
- If the system has updated, the test should suspend and resume after reviewing the new version
- If the project team gains enough funding, the testing should be resumed

5. Environmental Needs

5.1 System Hardware

We have the following requirements for the server of the back-end database needed for the test

- CPU main frequency more than 2.0GHZ
- Memory more than 2GB
- Hard disk capacity greater than 128G, hard disk speed more than 5400 rpm
- Network card: 100 megabit network card
- Network cable can be used normally and has good data transmission capability
- Power supply is sufficient
- Database using SQLite3

We have the following requirements for the client computer needed for the test

- CPU main frequency more than 2.0GHZ
- Memory more than 2GB

- Hard disk capacity more than 128G, hard disk speed more than 5400 rpm
- Network card: 100 megabit network card
- Network cable can be used normally, and has good data transmission
- capability Power supply sufficient power
- Mouse, keyboard, monitor, host is not damaged, can be used normally

5.2 Software Elements in the Test Environment

Type	Requirement
Operating System	Any major distribution, including Ubuntu, macOS, and Windows 10+
Browsers	All common browsers are available for testing, with Internet Explorer requiring
Database	SQlite version 3.0.3
Test Data	Designed by the test team

Table 1 Test Software Environment Table

6. Responsibilities, Staffing, and Training Needs

6.1 People and Roles

Test team members: All members of Software Engineering Management & Software Requirements Engineering G22, the specific test team responsibilities are arranged as shown in the following table

Role Name	Personnel Arrangement	Specific Responsibilities
Test Manager	Zhang Yichang	Supervise the whole project team, carry out the

		overall planning and coordination of the testing process, arrange each team member to carry out appropriate work, manage the testing progress, and be responsible for the development of test plans and the preparation of test reports.
Test Analysts	Zhang Qi, Nie Junzhe	Determining and defining the various tests and assignments to be performed, including determining test ideas, defining details, determining test results and documentation change requirements.
Test process designer	Kang Dakai	Define the methods and processes of testing, predict the effectiveness of testing, and evaluate the testing process.
Database Administrator	Kaihang Pan, Nan Li	Responsible for managing the database and providing database CRUDapi
Test sample designers	Qi Zhang, Junzhe Nie	Responsible for designing test cases, defining the operation of test classes, and supporting the test components required for the testability requirements defined by the test team
Testers	G22 All members	Responsible for conducting actual software system tests and recording feedback on test results.
Test machines	G22 Personal computers for all members	Personal laptops for team members

6.2 Staffing and Training Needs

- Test staffing:Zhang Yichang, Zhang Qi, Nie Junzhe, Pan Kaihang, Kang Dakai, Li Nan

- System training:Read the operating instructions and user manual of the online teaching system

- Training content:Learn the basic concepts of unit testing, stress testing, black box testing and white box testing, master the basic unit testing methods and various testing tools such as JEST and Jmeter.

7. Key Project/ Phase Milestones

Milestone	**Planned Start Date**	**Actual Start Date**	**Planned End Date**	**Actual End Date**
Project/ Phase starts	12.20	12.20	12.20	12.20
SQA Test Plan agreed	12.21	12.21	12.21	12.21
Testing resources requisitioned	12.21	12.21	12.21	12.21
Testing team training complete	12.21	12.21	12.21	12.21
Requirements baselined	12.21	12.21	12.21	12.21
Test Case Design baselined	12.22	12.22	12.22	12.22
QA – Cycle 1 Build Acceptance Test Execution	12.23	12.23	12.23	12.23
QA – Cycle 1 Functional Test Execution	12.24	12.24	12.24	12.24

QA – Cycle 2 Build Acceptance Test Execution	12.25	12.25	12.25	12.25
QA – Cycle 2 Functional Test Execution	12.26	12.26	12.26	12.26
QA Regression Test Execution	12.26	12.26	12.26	12.26
QA Performance/Failover Test Execution	12.27	12.27	12.27	12.27
QA Final Integrated Build Test Execution	12.28	12.28	12.28	12.28
Project Status Assessment review	12.29	12.29	12.29	12.29
Project/ Phase ends	12.30	12.30	12.30	12.30`

8. Risks, Dependencies, Assumptions, and Constraints

8.1 Risk

The possible risks of this test are as follows:

Possible Risks	Mitigation Strategies	Contingency Measures
Failure to meet the prerequisites for the start of the project	Testers need to meet certain prerequisites before the testing process can begin, and the system to be tested may meet certain conditions	Meeting the required prerequisites and considering load test failure<br/>
Inadequate preparation of test samples	Testers will prepare test data and test cases according to the required specifications	Test data and test cases are redefined, test plans are reviewed, and components are modified

Database needs to be refreshed	Ensure that the database manager regularly refreshes the database	restarts the system and database
Customer requirements are problematic	Test plans need to be redesigned, customer needs to define specific requirements	No risk of requirements, test plan failure
Insufficient time, there is a time risk	Test plan redesign	Test plan restarted
Testers not trained properly	Retrain	Replace a new group of testers

8.2 Dependency

- The team must go through a certain amount of training for the testers before conducting the testing work, which will affect the progress of the testing project
- The code development of the online teaching system must be completed on schedule before the testing work is carried out, which will affect the speed of test execution
- All the required resources must be ready before the testing work is carried out, which will affect the success of the test

All three of these dependencies are owned by the entire testing team.

8.3 Assumption

Proven assumptions	Impact of incorrect assumptions
Test team has been adequately trained	Test plan is delayed
Test resources and test samples are ready	Test plan failure
System development is completed and delivered on	Test plan cannot be

time with a high degree of requirements realization	carried out properly
Equipment is in good condition	Test cannot be carried out properly

8.4 Constraint

- Tests must be completed on time and before each milestone, and this constraint affects the progress of the testing effort
- All team members must participate and complete their work on time and on schedule