Technology Science nformation Networks Computing



Lecturer: Ting Wang (王挺)

利物浦大学计算机博士 清华大学计算机博士后 电子信息技术高级工程师 上海外国语大学网络与新媒体副教授 浙江清华长三角研究院海纳认知与智能研究中心主任

Jew Media Product Design and Development

Lecture 5. Testing

Dr. Ting WANG



School of Journalism and Communication
Shanghai International Studies University



Haina Cognition and Intelligence Research Center Yangtze Delta Region Institute of Tsinghua University, Zhejiang

Part 01 An overview to software testing

Testing in the software development



What is software testing?

Software Testing Definition according to ANSI/IEEE 1059 standard --

A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

https://www.softwaretestingmaterial.com/software-testing/

"Testing shows the presence, not the absence of bugs." —Edsger W. Dijkstra

Edsger W. Dijkstra (May 11, 1930 - Aug 06, 2002) was a Dutch systems scientist, programmer, software engineer, science essayist, and pioneer in computing science.



Bug

Fault (also called "defect" or "bug") is an erroneous hardware or software element of a system that can cause the system to fail.

Costs Time Development Unit Tests **QA** Testing Production

The longer a software bug exists throughout the product life-cycle, the more it costs. In 2002, software bugs cost the United States economy approximately \$59.5 billion. In 2016, that number jumped to \$1.1 trillion. https://dzone.com/articles/api-testing-best-practices

Reasons why software has bugs

human mistakes in software design and coding.

20 reasons for software bugs

Reasons in development

#1) Miscommunication or No Communication

- #2) Software Complexity
- #3) Programming Errors
- #4) Changing Requirements
- #5) Time Pressures
- #6) Egotistical or Overconfident People
- #7) Poorly Documented Code
- #8) Software Development Tools
- #9) Obsolete Automation Scripts

#10) Lack of Skilled Testers

Reasons in testing

#11) Not having a proper test setup (test environment) for testing all requirements.

#12) Starting to write code or test cases without understanding the requirements clearly .

#13) The incorrect design which leads to issues being carried out in all phases of the Software Development Cycle.

#14) Releasing software patches frequently without completing the Software Testing Life Cycle.

#15) Not providing training to resources for the skills needed for developing or testing the application properly.

#16) Giving very little or no time for Regression Testing.

#17) Not Automating Repetitive Test Cases and depending on the testers for manual verification every time.

#18) Not prioritizing test execution.

#19) Not tracking the development and test execution progress continuously. Last-minute changes are likely to introduce errors.

#20) The wrong assumption made while coding and testing stages.

Why testing?

humans make mistakes all the time

- 1. Cost-effectiveness
- 2. Customer Satisfaction
- 3. Security
- 4. Product Quality

Objectives

- 1. to evaluate the functionality of a software application
- 2. to find whether the developed software met the specified requirements or not
- 3. to identify the defects to ensure that the product is defect-free in order to produce the quality product
- 4. to find faults as cheaply and quickly as possible



Why software testing is hard?

Ideally, we would design a single "right" test case to expose each fault and run it. However, any nontrivial system cannot be completely tested.

Test case are used for software testing, the more test cases, the higher the cost. Test case and test scenario

Test case : How to test

a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test scenario : What to test

a statement describing the functionality of the application to be tested.

https://artoftesting.com/test-scenario-examples



https://artoftesting.com/test-case



Example: write a test case

| Test case ID | Test case description | Prerequisites | Test steps | Test data | Expected Result | Actual Result | Status | Created By | Date of creation | Executed By | Date of execution |
|--------------|------------------------|-----------------------|----------------------------|-----------|-------------------|-----------------|--------|------------|------------------|-------------|-------------------|
| TC001 | The objective of this | 1. User is authorized | 1. Enter valid username | 1. User | 1. User should | 1. If the valid | Fail | Rajesh | 1/1/2016 | Umesh | 1/2/2016 |
| | test case is to verify | 2. Has an account in | 2. Enter valid password | account | be able to login | credentials | | | | | |
| | the 'Login' of Gmail | Gmail | 3. Click on 'Login' button | should | his Gmail | are entered | | | | | |
| | account | | | be | account with his | then the user | | | | | |
| | | | | present | valid credentials | will be able | | | | | |
| | | | | in Gmail | | to login his / | | | | | |
| | | | | | 2. 'Invalid | her account | | | | | |
| | | | | | username or | | | | | | |
| | | | | | password' | 2. If invalid | | | | | |
| | | | | | should get | credentials | | | | | |
| | | | | | displayed if the | are entered | | | | | |
| | | | | | username and | then nothing | | | | | |
| | | | | | password are | happens(the | | | | | |
| | | | | | not valid | expected | | | | | |
| | | | | | | message is | | | | | |
| | | | | | | not | | | | | |
| | | | | | | displayed) | | | | | |

Test case ID: The ID of the test case

Test case description: The objective or summary of the test case **Prerequisites:** Any preconditions which need to be executed before starting the test

Test steps: Procedure to execute the test

Test data: Data required while executing the test **Expected Result:** The expected output of the test **Actual Result:** The actual output of the test

Status: Pass, Fail, 'Not executed' when test case is not executed and 'Blocked' when high severity bug is found Created By: Name of the person who wrote the test case Date of creation: The date on which the test case was authored Executed By: Name of the person who ran or executed the test case Date of execution: The date on which the test case was executed https://www.wikihow.com/Write-a-Test-Case http://tryga.com/test-case/ Types of software testing

Manual Testing

testing software by hand

https://www.softwaretestingmaterial.com/manual-testing-tutorial/

Automation Testing

testing the software using an automation tool

| BEST AUTOMATION TESTING TOOLS | | | | | | | | |
|-------------------------------|------------------------------------|------------------------------------|--|--|--|--|--|--|
| | © www.SoftwareTestingHelp.com | | | | | | | |
| 1. Ranorex | 7. LambdaTest | 13. Selenium | | | | | | |
| 2. TestComplete | 8. Qualibrate | 14. Subject7 | | | | | | |
| 3. QMetry Automation Studio | 9. Worksoft | 15. Appium | | | | | | |
| 4. TestProject | 10. ZeuZ Test Automation Framework | 16. Micro Focus UFT | | | | | | |
| 5. Katalon Studio | 11. CrossBrowserTesting | 17. Test Studio | | | | | | |
| 6. Testsigma | 12. Testimony | 18. IBM Rational Functional Tester | | | | | | |

Testing Methods

Static Testing

Verification to check documents and files whether we are developing the product accordingly or not

Dynamic Testing

Validation to test the real product whether we are building the right product or not

VERIFICATION VS VALIDATION \bigcirc Verification is the process, Validation is the process, to ensure that whether we whether we are building the are building the product right product i.e., to validate right i.e., to verify the the product which we have requirements which we developed is right or not. have and to verify whether we are developing the product accordingly or not. As per IEEE-STD-610: The As per IEEE-STD-610: The process of evaluation process of evaluating software to determine software during or at the whether the products of end of the development a given development process to determine phase satisfy the whether it satisfies conditions imposed at specified requirements the beginning of that [IEEE-STD-610] phase. It answers the question. It answers the question. Am I building a product Am I building a right right? product? Activities involved in this Activities involved here is Testing the software are Inspections, Reviews, application by means of Walkthroughs White box. Grav box. & Black box testing. It's a Low-Level Activity. It's a High Level Activity. It is a static method of It is a dynamic process of checking documents and testing the real product. files. It doesn't involve code It involves code execution execution Low cost compared to Costly compared to validation tests verification tests

Testing Approaches



White Box Testing

 also called as Glass Box, Clear Box, Structural Testing. White Box Testing is based on application's internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.

OPENIUMO UGE Blac • also sof

Black Box Testing

 also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.



Grey Box Testing

 the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

Testing Levels



Unit Testing



Integration Testing

System Testing



Acceptance Testing

Logical Organization of Testing



Testing Artifacts

What are testing artifacts?

the deliverables that are given to the stakeholders of a software project, some of the deliverables are provided before the testing phase commences and some are provided during the testing phase and rest after the testing phase is completed.

Some of the test deliverables are as follows:

- Test plan
- Traceability matrix
- Test case
- Test script
- Test suite
- Test data or Test Fixture
- Test harness



Principles of Software Testing

- 1. Testing shows the presence of defects
- 2. Exhaustive testing is impossible
- 3. Early testing
- 4. Defect clustering
- 5. Pesticide paradox
- 6. Testing is context-dependent
- 7. Absence of error fallacy



GB/T8567

1988

测试计划 测试分析报告

2006

软件测试计划(STP) 软件测试说明(STD) 软件测试报告(STR)





Dr. Ting WANG



School of Journalism and Communication
Shanghai International Studies University



Haina Cognition and Intelligence Research Center Yangtze Delta Region Institute of Tsinghua University, Zhejiang