Fundamentals of Testing

Presentation Outline

- *Software Errors*
- *Software Test Planing*
- *Software Test Requirements*
- *Software Test Design*
- *Review for Software Testing*
- *Software Test Execution*
- *Problem Tracking System*
- *Software Test Management*
- *Software Test Engineers' Tasks*
- *Test Automation and Tools*

Categories of Software Errors

- *User interface errors, such as output errors, incorrect user messages.*
- *Function errors*
- *Incorrect program version*
- *Requirements errors*
- *Documentation errors*
- *Module interface errors*
- *Error handling*
- *Logic errors, such as calculation errors*
- *State-based behavior errors*
- *Program structure errors, such as control-flow errors*
- *Defect hardware*
- *Testing errors*
- *Design errors*
- *Architecture errors*
- *Performance errors*
- *Boundary-related errors*
- *Communication errors*

Software Errors

What is a software error?

One common definition of a software error is a mismatch between the program and its specification.

Definition #1:

"A mismatch between the program and its specification is an error in the program if and only if the specification exists and is correct."

Definition #2:

"A software error is present for when the program does not do what its end user reasonability expects to do." (Myers, 1976)

Definition #3:

"There can never be an absolute definition for bugs, nor an absolute determination of their existence. The extent to which a program has bugs is measured by the extent to which it fails to be useful. This is a fundamentally human measure." (Besizer, 1984)

Software Test Planing

Like other activities in software engineering phases, it is impossible to have a cost-effective software test process without a very good planing,

The major objective of software test planing:
- generate a well-defined software test plan.

What content should be included in a software test plan?

- ***Testing activities and schedule***
- ***Testing tasks and assignments***
- ***Selected test strategy and test models***
- ***Test methods and criteria***
- ***Required test tools and environment***
- ***Problem tracking and reporting***
- ***Test cost estimation***

Other needed items: ***quality control process and standards***

Software Test Requirements

Before starting test design, we must identify our test objectives, focuses, and test items. The major purpose is to help us understand what are the targets of software testing.

This step can be done based on:

- ***Requirements specifications***
- ***Inputs from developers***
- ***Feedback from customers***

Benefits are:

- ***Identify and rank the major focus of software testing***
- ***Check the requirements to see if they are correct, completed, and testable***
- ***Enhance and update system requirements to make sure they are testable***
- ***Support the decision on selecting or defining test strategy***

For example,

- for performance testing, we need clear requirements on system performance.

Software Test Requirements

The essentials of testing requirements include:

- ***Specified testing methods***
- ***Required test types and test coverage criteria***
- ***Selected or required test tools***
- ***Testing focuses and test items for each type of software testing***

An example of performance testing requirements:

> ***"Check the system performance to make sure it meet 99% system reliability requirements"***

A typical example for required test items is:

Test item #I: "Test the call waiting feature (REQ #j) during system testing based on the given requirements specifications."

Software Test Design

Software test design is an important task for software test engineers.

A good test engineer always know how to come out quality test cases and perform effective tests to uncover as many as bugs in a very tight schedule.

What do you need to come out an effective test set ?

- Choose a good test model and an effective testing method
- Apply a well-defined test criteria
- Generate a cost-effective test set based on the selected test criteria
- Write a good test case specification document

What is a good test case?

- ***It must have a high probability to discover a software error***
- ***It is designed to aim at a specific test requirement***
- ***It is generated by following an effective test method***
- ***It must be well documented and easily tracked***
- ***It is easy to be performed and simple to spot the expected results***
- ***It avoids the redundancy of test cases***

Software Test Design

What content should be included in a test case?.

Test Case ID:	***Test Item:***
Wrote By: (tester name)	***Documented Date:***
Test Type:	***Test Suite#:***
Product Name:	***Release and Version No.:***

Test case description:

Operation procedure:

Pre-conditions:	***Post-conditions:***
Inputs data and/or events:	***Expected output data and/or events:***

Required test scripts:

Software Test Execution

Test execution can be performed:

- ***using manual approach***
- ***using a systematic approach***
- ***using a semi-automatic approach***

Basis activities in test execution are:

- ***Select a test case***
- ***Set up the pre-conditions for a test case***
- ***Set up test data***
- ***Run a test case following its procedure***
- ***Track the test operations and results***
- ***Monitor the post-conditions of a test case & expect the test results***
- ***Verify the test results and report the problems if there is any***
- ***Record each test execution***

Software Test Execution

What do you need to perform test execution?

- ***a test plan***
- ***test design specification with test case specifications***
- ***a test suite with documented test cases (optional)***
- ***test supporting facility, such as test drivers, test stubs, simulators***

What are the outcome of an test execution:

- ***Text execution record and report***
- ***Problem and bug reports***
- ***Error logs***

With automatic test execution tools (or test control tools), we can do:

- ***automatic test runner***
- ***record and replay***

Problem Analysis and Report

When do we issue a problem?

Whenever a bug or problem is found, we need to write down a problem report immediately.

What are the content of a problem report?

Problem ID	***current software name,***	***release no. and version no.***	
Test type	***Reported by***	***Reported date***	***Test case ID***
Subsystem (or module name)		***Feature Name (or Subject)***	
Problem type (REQ, Design, Coding, …)		***Problem severity (Fatal, Major, Minor, ..)***	
Problem summary and detailed description			
Cause analysis	***How to reproduce?***	***Attachments***	

Problem Analysis and Report

How to track, control, and manage issued problems?

- A systematic solution is needed to track and maintain the reported problems in a repository.
- Define and implement a problem control and analysis process to control problem tracking, reporting, analysis, fixing and resolutions.

Characteristics of a problem report:

- Simple and understandable

- Traceable and numbered

- Reproducible

- Non-judgmental

Problem analysis:

-Finding the most serious consequences
-Finding the simplest and most general conditions
-Finding alternative paths to the same problem
-Finding related problems

Software Test Review

What is a review?

- A review is a verification activity to assure a correct method has been used to create a software product.

- Participants in a review take full responsibility for results.

There are two types of reviews:

- ***Formal reviews:***
  - ***use a well-defined review method (or technique)***
  - ***generate formal review results***

- ***Informal reviews***
  - ***use a desk checking approach***
  - ***conduct an informal review***
  - ***generate information review results***

Software Test Review

Products should be reviewed during software testing:

Test Plan	***Test Design Specification***
Test Procedure Specification	
Test Report	***Problem Reports***

What do reviews accomplish?

- Reviews provide the primary mechanism for reliably evaluating progress.

- Reviews train and educate the participants and have a significant positive effect on staff competence.

- Reviews give early feedback and prevent more serious problems from arising.

- Reviews bring individual capability to light.

Test Management

Test management encompass:

- ***management of a test team***
- ***management of a test process***
- ***management of test projects***

A test manager's role:

- Play as a leadership role in:

- ***planing projects*** ***- setting up a direction*** ***-motivating people***
- ***build a team*** ***- manage engineers***

- Play as a controller in:

- ***product evaluation*** ***- performance evaluation***
- ***changing to a new direction***

- Play as a supporter in:

- ***assist and train engineers*** ***- train engineers***
- ***enforce and control test methods, standards, and criteria***
- ***select and develop test tools***

Test Management

Test management functions:

- Management	***- Manage test projects***
	- Manage team members
	- Manage test processes
- Motivation	***- Motivate quality work from team members***
	- Simulate for new ideas and creative solutions
- Methodology	***- Control of setting up test methodology, process, standards.***
	- Control of establishing test criteria
- Mechanization	***- Control the selection and development of test tools***
	- Mechanism for the configuration management of test suites
	- Control of setting up an integrated test environment
- Measurement	***- Measure test cost, complexity and efforts***
	- Measure engineer performance
	- Measure test effectiveness
	- Measure product quality

Test Management

Test management encompass:

- ***management of a test team***
- ***management of a test process***
- ***management of test projects***

A test manager's role:

- Play as a leadership role in:

- ***planing projects*** ***- setting up a direction*** ***-motivating people***
- ***build a team*** ***- manage engineers***

- Play as a controller in:

- ***product evaluation*** ***- performance evaluation***
- ***changing to a new direction***

- Play as a supporter in:

- ***assist and train engineers*** ***- train engineers***
- ***enforce and control test methods, standards, and criteria***
- ***select and develop test tools***

Test Management

Questions for managers who claim to be managing software testing:

Management:	***Do you plans address testing?*** ***Do you know who is responsible?*** ***Have you published your testing policy?***
Motivation:	***Do you provide incentive for people do quality work?*** ***Do you encourage people to take advantage of training opportunities in testing methods?***
Methodology:	***Are your testing methods proceduralized and are people trained in their use?*** ***Are you aware of new testing techniques and are you working to introduce them?***

Test Management

Questions for managers who claim to be managing software testing:

Mechanization:	***Do you sufficient hardware and equipment to support testing?***
	Have you provided appropriate software testing tools and aids?
	Do you evaluate automated testing aids on an ongoing basis?
Measurement:	***Do you track errors, faults, and failures?***
	Do you know what testing costs?
	Do you quantitatively measure testing performance?

Test Engineers' Tasks

What does a test engineer do?

- Ensure that testing is performed
- Ensure that testing is documented
- Ensure that testing methodology, techniques, standards are established and developed

The basic skills for a qualified test engineer:

Controlled	***- Organized individual, systematic planing***
	- Good planing on testing
Competent	***- Technical awareness of testing methods, tools, and criteria***
Critical	***- Inner determination to discover problems***
Comprehensive	***- Total understanding of the given system and specifications***
	- Pay good attention to the details
Considerate	***- Ability to related to others and resolve conflicts***

Basic Test Engineers' Tasks

The basic tasks of a test engineer include:

- ***Prepare testing plans and organize testing activities***
- ***Design test cases and document them using well-defined test methods***
- ***Develop test procedures and test data***
- ***Write problem reports and text execution records***
- ***Use testing tools and aids***
- ***Review test designs, results and problems***
- ***Test maintenance changes***
- ***Oversee acceptance tests***

Test Automation and Tools

What is software test automation?

Software test automation refers to a process and activities that reduce manual testing activities and efforts in a software development lifecycle.

Why software test automation?

- Reduce software testing time, cost, and efforts
- Increase the quality of software testing
- Apply well-defined test methods through tools
- Relieve the complicated and redundant work from test engineers

What do we need to automate software testing?

- Limited cost and schedule
- Select and introduce quality test tools
- Develop necessary effective and efficient test tools and facility
- Apply well-defined testing methods and coverage
- Form an integrated test environment supporting various software testing activities for products on a production line

Test Automation and Tools

Basic steps for test automation:

Level #1: Automatic test management
- ***test case & suite management, and documentation***
- ***test script management***

Level #2: Automatic test execution
- ***black-box test control and execution***
- ***white-box test control and execution***

Level #3: Automatic test generation
- ***black-box test generation***
- ***white-box test generation***

Level #4: Automatic test measurement
- ***test coverage and metrics measurement***
- ***test cost and complexity measurement***

Test Automation and Tools

Classification of software test tools:

- GUI record and replay tools
- Program specification-based test tools

- Test management and configuration management tools
- Test generation tools, such as random test tools

- Program-based test tools (or white-box test tools)
- Program test metrics tools

- Program performance test tools
    - performance monitoring tools
    - performance evaluation tools
    - Program load test tools

- Protocol-based confirmation tool tools
- Program regression test tools (such as program change analysis tool)