

MODULE #2 - SOFTWARE TESTING FUNDAMENTALS

Topic #2 – Software Test Design

Instructor: Jerry Gao, Ph.D., Professor San Jose State University







Software Test Design Basics

Software Test Design Principles

Software Test Case Templates

Software Testing Myths

Software Testing Limitations





Software Test Case Design Basics - I

Software test design is an important task for test engineers.

A good test engineer always know:

- → How to come out quality test cases and
- → How to 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





Software Test Case Design Basics-II

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 Case Template

What content should be included in a test case?

Test Case ID: Test Item:

Wrote By:(tester name) Documented Date:

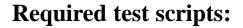
Test Type: Test Suite#:

Product Name: Release & Version No.:

Test case description: Operation procedure:

Pre-conditions: Post-conditions:

Inputs data and/or events: Expected output data & events:







Software Test Design Principles

•Principle #1: Complete testing is impossible.

•Principle #2: Software testing is not simple.

•Principle #3: Testing is risk-based.

•Principle #4: Testing must be planned.

•Principle #5: Testing requires independence.

•Principle #6: Quality software testing depends on:

- Good understanding of software products and related domain application
- O Cost-effective testing methodology, coverage, test methods, and tools.
- o Good engineers with creativity, and solid software testing experience







Software Testing Myths

- We can test a program completely. In short, we must test a program exhaustively.
- We can find all program errors as long as test engineers do a good job.
- We can test a program by trying all possible inputs and states of a program.
- A good test suite must include a great number of test cases.
- Good test cases always are complicated ones.
- Test automation can replace test engineers to perform good software testing.
- Software testing is simple and easy. Anyone can do it. No training is needed.







Software Testing Limits

- Due to the testing time limit, it is impossible to achieve total confidence.
- We can never be sure the specifications are 100% correct.
- We can never be certain that a testing system (or tool) is correct.
- No testing tools can copy with every software program.
- Test engineers never be sure that they completely understand a software product.
- We never have enough resources to perform software testing.
- We can never be certain that we achieve 100% adequate software testing.

