

Introduction to Software Testing

Speaker: Jerry Gao Ph.D.

***San Jose State University
email: jerrygao@email.sjsu.edu
URL: <http://www.engr.sjsu.edu/gaojerry>***

Presentation Outline

- ***What is Software Testing?***
 - ***Definitions***
 - ***Testing Objectives***
 - ***Who Does Software Testing?***
- ***Software Testing Activities***
- ***Software Testing Scope***
- ***Software Testing Principles***
- ***Software Testing Process***
- ***Software Testing Myths***
- ***Software Testing Limits***
- ***Different Types of Software Testing***

What is Software Testing

Several definitions:

“Testing is the process of establishing confidence that a program or system does what it is supposed to.” by Hetzel 1973

*“Testing is the process of executing a program or system with the intent of finding errors.”
by Myers 1979*

*“Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.”
by Hetzel 1983*

Testing Objectives

The Major Objectives of Software Testing:

- *Uncover as many as errors (or bugs) as possible in a given timeline.*
- *Demonstrate a given software product matching its requirement specifications.*
- *Validate the quality of a software testing exercise using minimum cost and efforts.*
- *Generate high quality test cases, perform effective tests, and issue correct and helpful problem reports.*

Major goals:

uncover the errors (defects) in the software, including errors in:

- *requirements from requirement analysis*
- *design documented in design specifications*
- *coding (implementation)*
- *system resources and system environment*
- *hardware problems and their interfaces to software*

Who does Software Testing

- *Test manager*
 - *manage and control a software test project*
 - *supervise test engineers*
 - *define and specify a test plan*

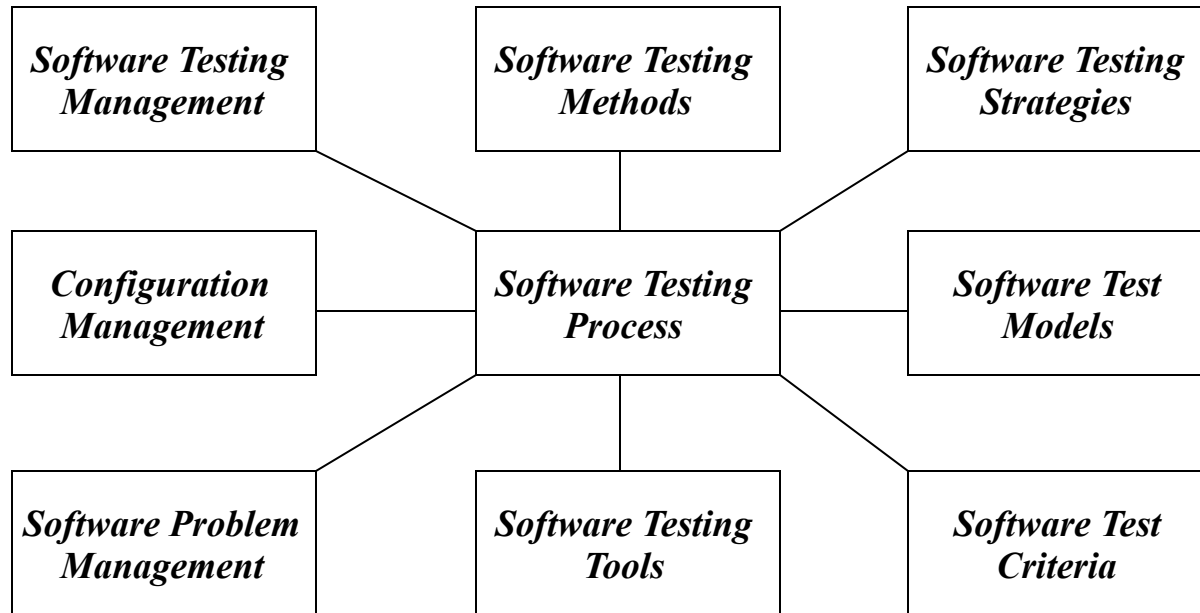
- *Software Test Engineers and Testers*
 - *define test cases, write test specifications, run tests*

- *Independent Test Group*

- *Development Engineers*
 - *Only perform unit tests and integration tests*

- *Quality Assurance Group and Engineers*
 - *Perform system testing*
 - *Define software testing standards and quality control process*

Software Testing Scope



Software Testing Activities

Generic characteristics of software testing strategies:-

- *Testing begins at module level and works outward towards the integration of entire computer based system.*
 - *Different testing techniques are required at different points in time.*
- *Testing is conducted by the s/w developer and ITG (Independent Test Group) for large projects.*
- *Testing and Debugging are different and Debugging is essential in any testing strategy.*

Software Testing Activities

- Test Planing

Define a software test plan by specifying:

- a test schedule for a test process and its activities, as well as assignments***
- test requirements and items***
- test strategy and supporting tools***

- Test Design and Specification

- Conduct software design based well-defined test generation methods.***
- Specify test cases to achieve a targeted test coverage.***

- Test Set up:

- Testing Tools and Environment Set-up***
- Test Suite Set-up***

- Test Operation and Execution

- Run test cases manually or automatically***

Software Testing Activities

- *Test Result Analysis and Reporting*
Report software testing results and conduct test result analysis
- *Problem Reporting*
Report program errors using a systematic solution.
- *Test Management and Measurement*
Manage software testing activities, control testing schedule, measure testing complexity and cost
- *Test Automation*
 - *Define and develop software test tools*
 - *Adopt and use software test tools*
 - *Write software test scripts and facility*
- *Test Configuration Management*
 - *Manage and maintain different versions of software test suites, test environment and tools, and documents for various product versions.*

Verification and Validation

*Software testing is one element of a broader topic that is often referred to as
====> Verification and Validation (V&V)*

Verification --> refers to the set of activities that ensure that software correctly implements a specific function.

Validation -> refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

Boehm [BOE81]:

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

The definition of V&V encompasses many of SQA activities, including formal technical reviews, quality and configuration audits performance monitoring, different types of software testing feasibility study and simulation

Software Quality Factors

Functionality (exterior quality)

- *Correctness, reliability, usability, and integrity*

Engineering (interior quality)

- *Efficiency, testability, documentation, structure*

Adaptability (future qualities)

- *Flexibility, reusability, maintainability*

Software Testing Principles

- *Principle #1: Complete testing is impossible.*
- *Principle #2: Software testing is not simple.*
 - *Reasons:*
 - *Quality testing requires testers to understand a system/product completely*
 - *Quality testing needs adequate test set, and efficient testing methods*
 - *A very tight schedule and lack of test tools.*
- *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*
 - *Cost-effective testing methodology, coverage, test methods, and tools.*
 - *Good engineers with creativity, and solid software testing experience*

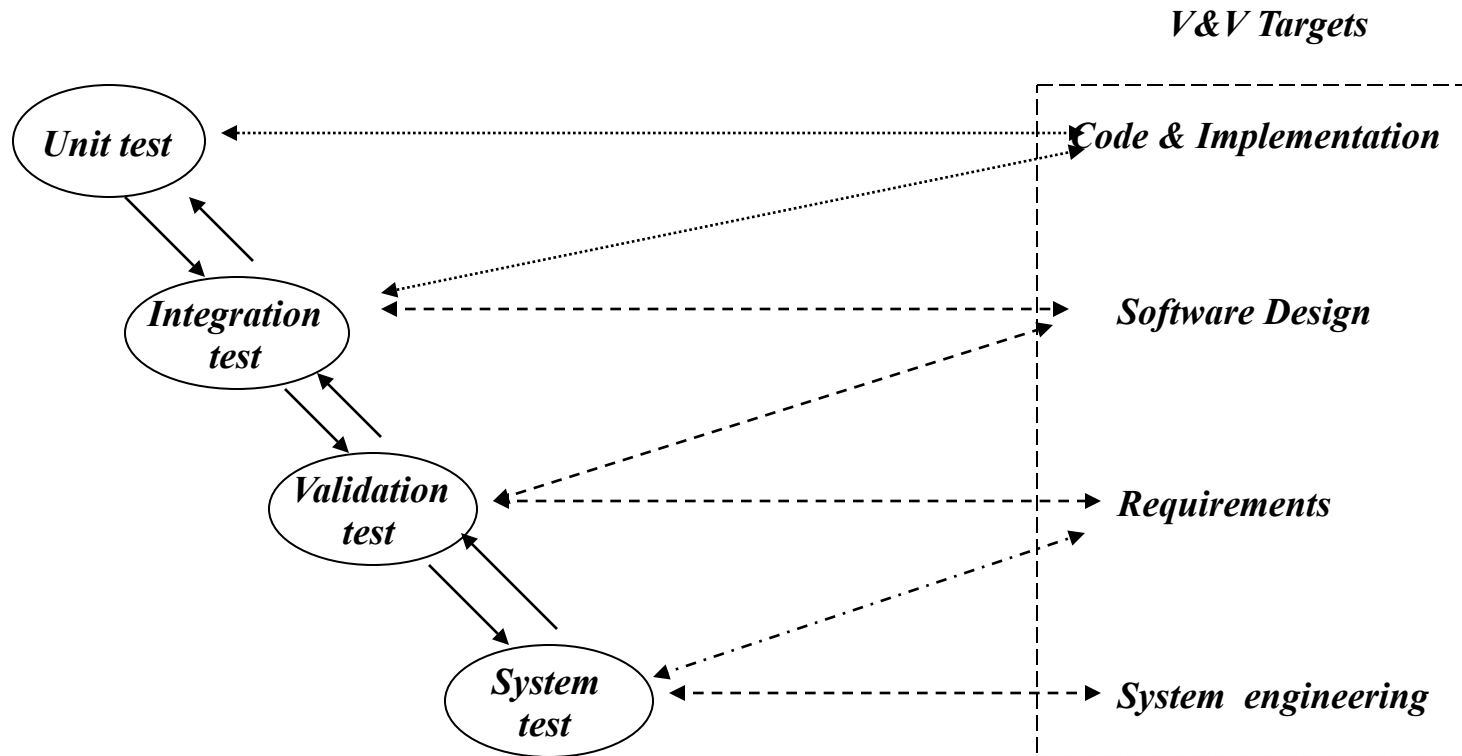
Software Testing Myths

- *We can test a program completely. In other words, we 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.*
- *Software 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 cope with every software program.*
- *Tester engineers can 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 have achieved 100% adequate software testing.*

Software Testing Process



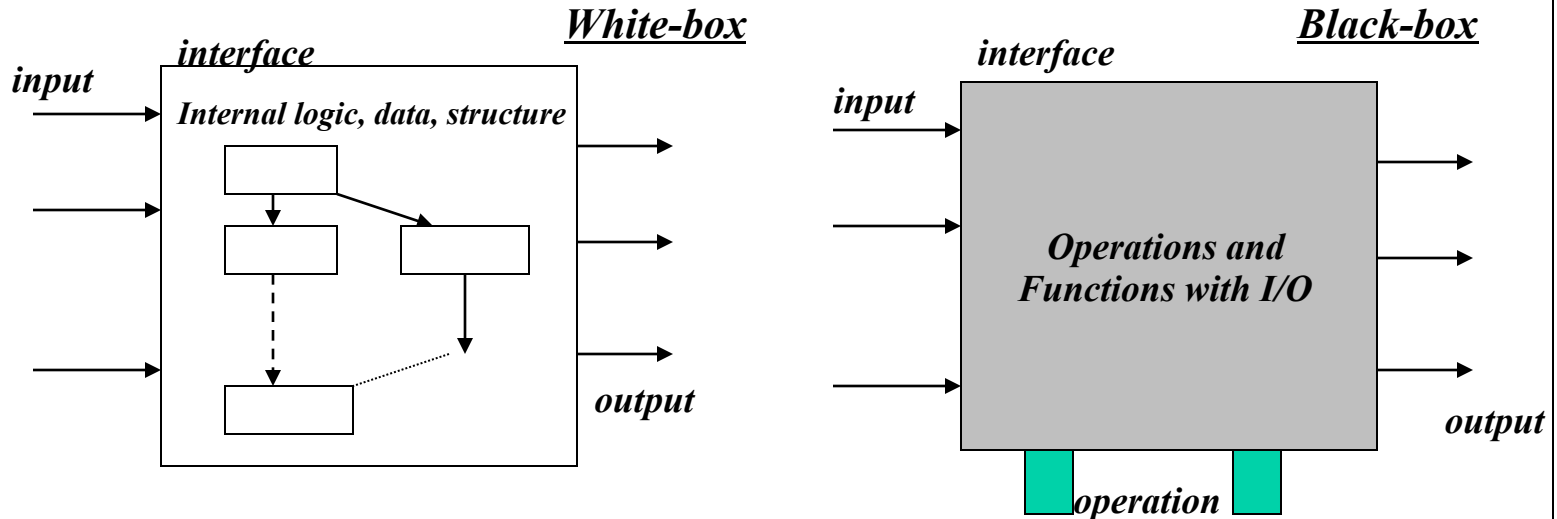
Unit Test (Component Level Test)

Unit testing: Individual components are tested independently to ensure their quality. The focus is to uncover errors in design and implementation, including

- data structure in a component
- program logic and program structure in a component
- component interface
- functions and operations of a component
- boundary conditions, independent paths and error handling

paths

Unit testers: developers of the components.



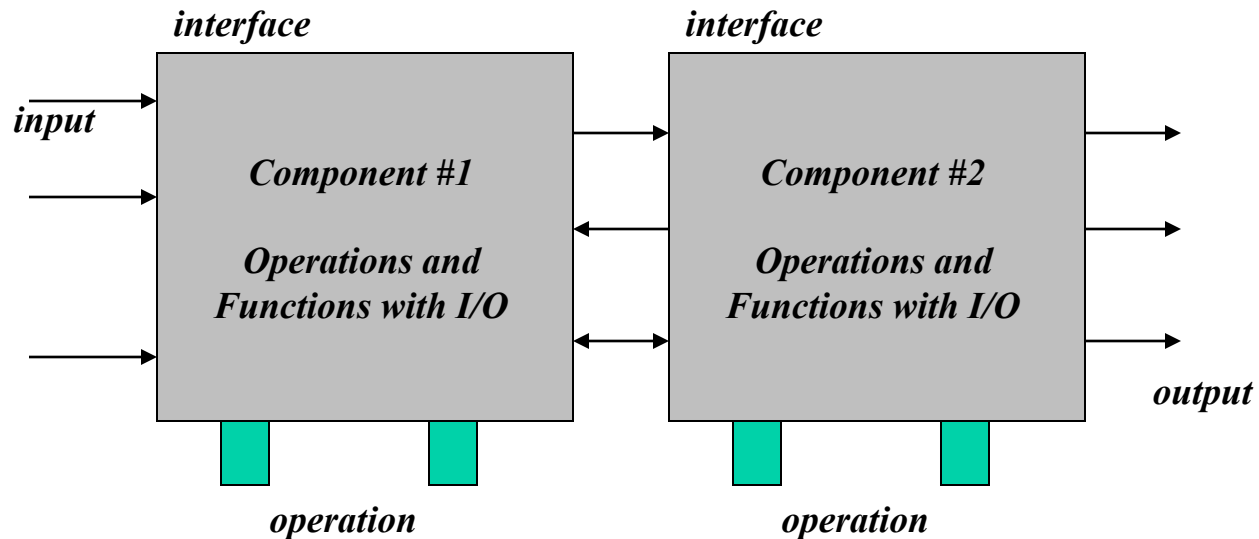
Integration Testing

Integration test: *A group of dependent components are tested together to ensure their the quality of their integration unit.*

The focus is to uncover errors in:

- *Design and construction of software architecture*
- *Integrated functions or operations at sub-system level*
- *Interfaces and interactions between them*
- *Resource integration and/or environment integration*

Integration testers: *either developers and/or test engineers.*



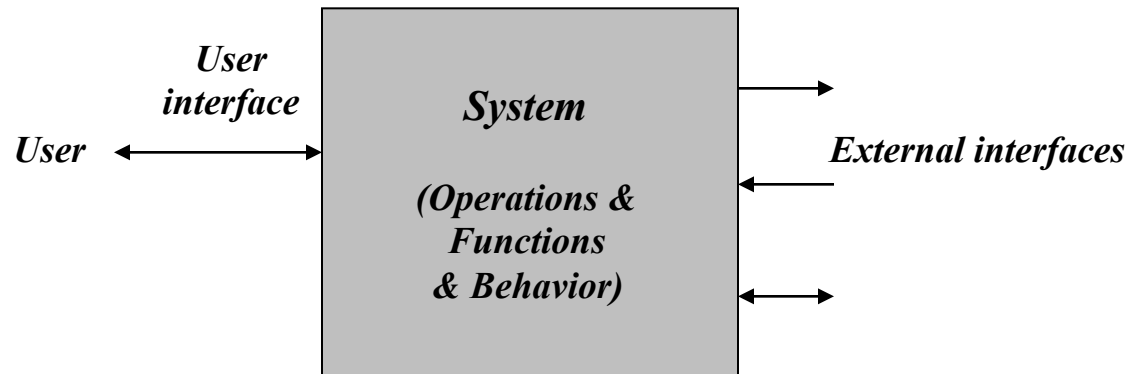
Function Validation Testing

Validation test: *The integrated software is tested based on requirements to ensure that we have a right product.*

The focus is to uncover errors in:

- *System input/output*
- *System functions and information data*
- *System interfaces with external parts*
- *User interfaces*
- *System behavior and performance*

Validation testers: *test engineers in ITG or SQA people.*



System Testing

System test: The system software is tested as a whole. It verifies all elements mesh properly to make sure that all system functions and performance are achieved in the target environment.

The focus areas are:

- *System functions and performance*
- *System reliability and recoverability (recovery test)*
- *System installation (installation test)*
- *System behavior in the special conditions (stress and load test)*
- *System user operations (acceptance test/alpha test)*
- *Hardware and software integration and collaboration*
- *Integration of external software and the system*

System testers: test engineers in ITG or SQA people.

When a system is to be marketed as a software product, a testing process called beta testing is often used.

Summary of Test Types

Unit testing

Exercises specific paths in a component's control structure to ensure complete coverage and maximum error detection

Components are then assembled and integrated

Integration testing

Focuses on inputs and outputs, and how well the components fit together and work together

Validation testing

Provides final assurance that the software meets all functional, behavioral, and performance requirements

System testing

Verifies that all system elements (software, hardware, people, databases) mesh properly and that overall system function and performance is achieved

Test Issues in Real World

Software testing is very expensive.

How to achieve test automation?????

When should we stop software testing?

Test criteria, test coverage, adequate testing.

Other software testing:

GUI Testing

Object-Oriented Software Testing

Component Testing and Component-based Software Testing

Domain-specific Feature Testing

Testing Web-based Systems

Helpful Tips

- One approach is to divide the test results into various severity levels
 - Then consider testing to be complete when certain levels of errors no longer occur or have been repaired or eliminated
- Understand the user of the software (through use cases) and develop a profile for each user category
- Develop a testing plan that emphasizes rapid cycle testing to get quick feedback to control quality levels and adjust the test strategy
- Build robust software that is designed to test itself and can diagnose certain kinds of errors
- Conduct formal technical reviews to assess the test strategy and test cases themselves

Glossary

Test Case:

Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc. set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Plan:

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.