

Software Testing Basics

After this course, students can have:

- Basic knowledge about software testing
- Basic concepts in software testing

❖ Course Duration: 4h

1. Test Definition
2. Test Process
3. Test Levels
4. Test Types
5. Test Techniques
6. Test Documents
7. Defect Concepts
8. Test Measurements
9. Tester Attitudes and Skills

1. Test Definition

What is Testing ?

- What is Testing ?

The process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements - (IEEE 83a)

- Test Objectives:

- *Primary*: Execute a program with the intent of finding defects to:

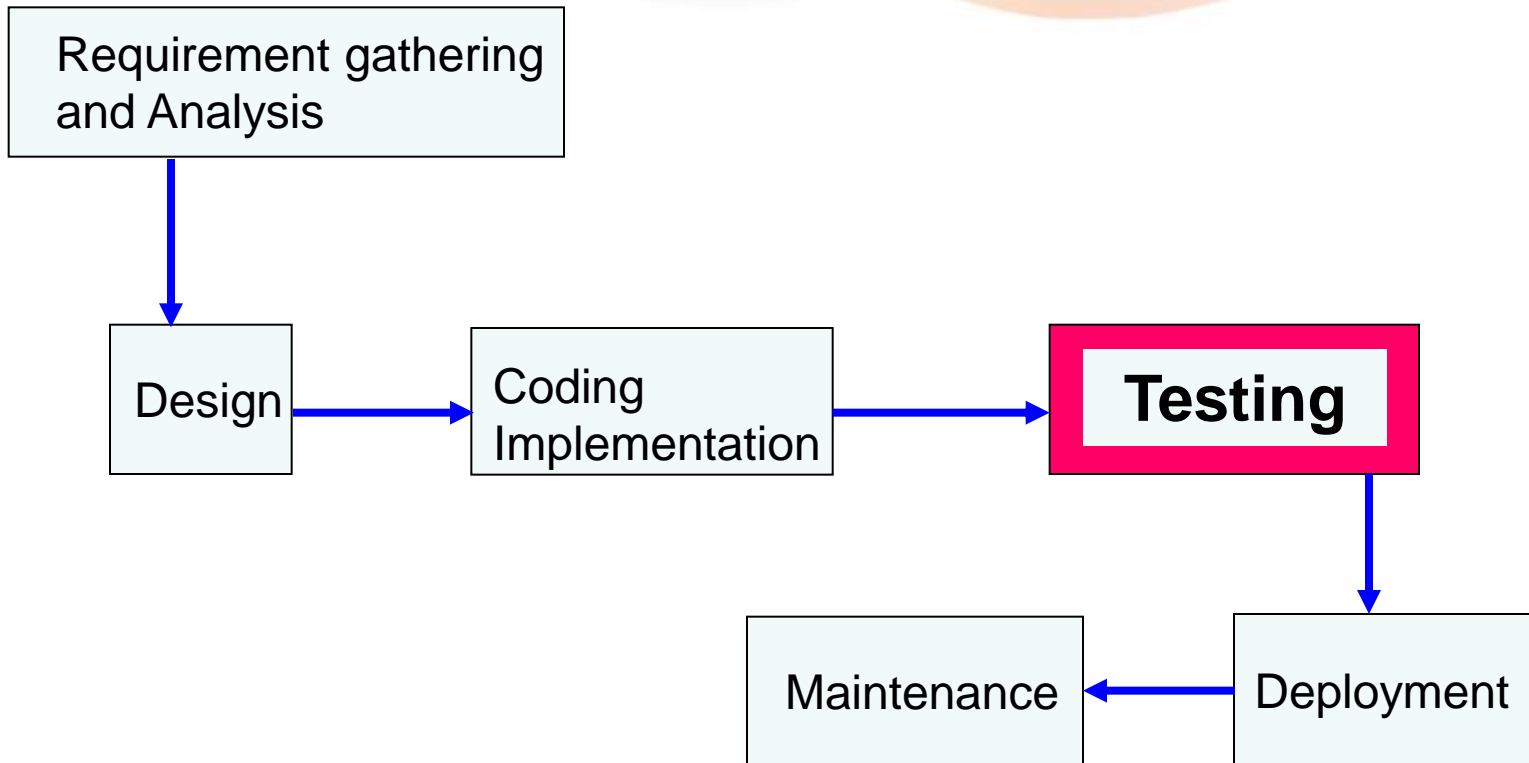
- Determine whether system meets specifications
 - Determine whether system meets user's needs

- *Secondary*: gain confidence about the level of quality, providing information, continuous improve test process

2. Test Process

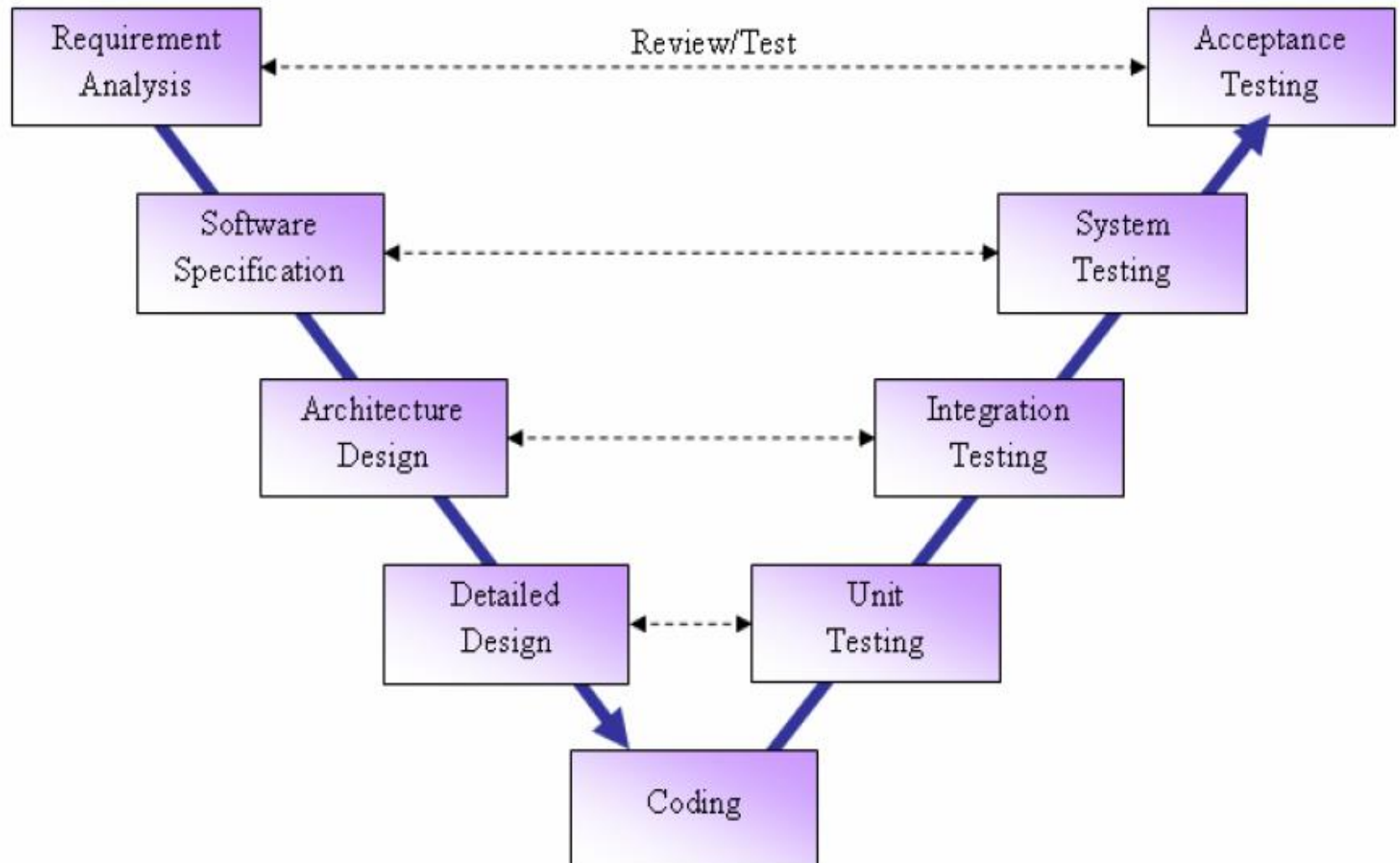
Testing within a life cycle model

- 6 phases in every SDLC:

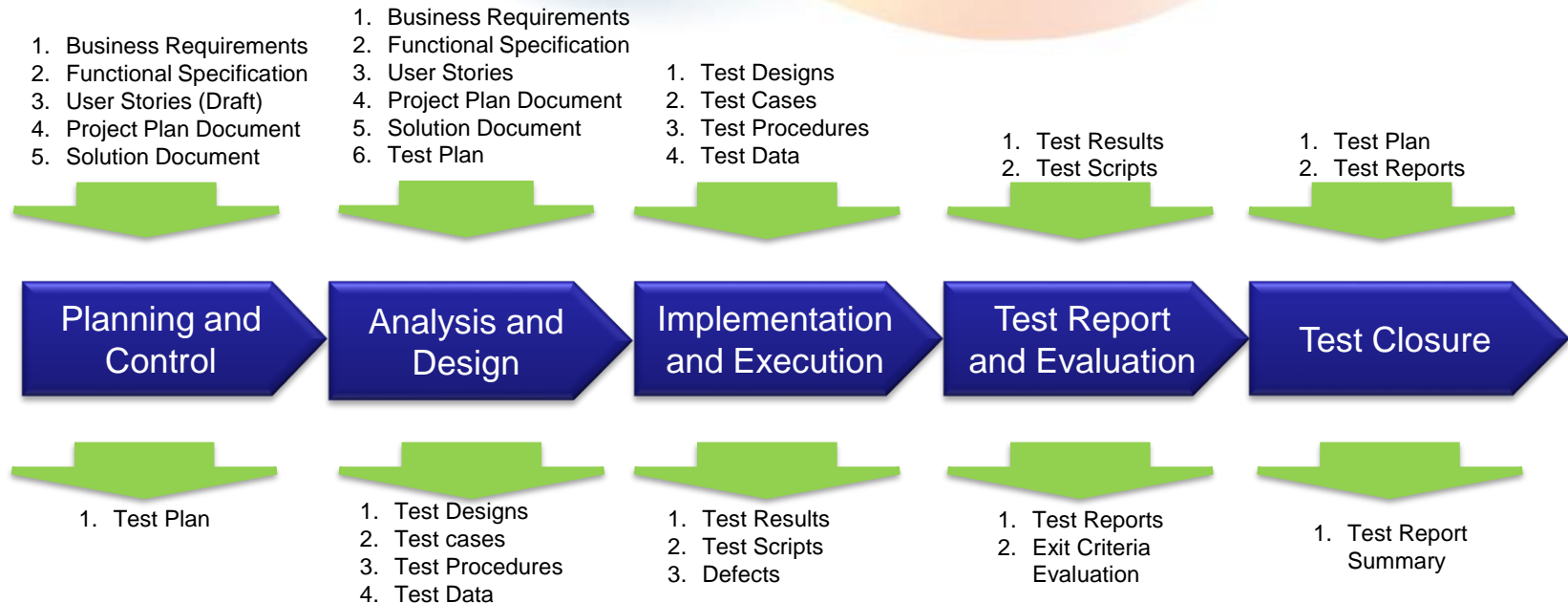


1. Testing Definition

Where are testing tasks?



2. Test Process



2. Test Process

Common Inputs and Outputs

- Test Planning:
 - Input: Project plan, Customer Requirement & Acceptance criteria/SRS
 - Output: Test Plan document
- Test Design:
 - Input: Test plan, Requirement, detail design
 - Output: Test cases, test scripts, test data in high level
- Test Implementation and Execution:
 - Input: Test cases, test data, test scripts
 - Output: Test report, Defect list

2. Test Process

Common Test Resources

- Guideline: Test process
- Templates for test documents:
 - Test Plan
 - Test case specification
 - Test report
 - Defect analysis report
- Test tools:
 - Defect tracking tool
 - Test Effort tracking tool
 - Test schedule
 - Test automation tools
 - *Rational Robot (Functional & Performance test)*
 - *OpenSTA (Open source), Wtir (Open source)*

3. Test Levels

- There are 4 levels of testing:
 - Unit Test (Component Test)
 - Integration Test
 - System Test
 - Acceptance Test



3. Test Levels

Unit Test/Component Test

- Component test:
 - The testing of individual software components (e.g. modules, programs, objects, classes, etc.) that are separately testable.
- Test cases are derived from work products:
 - Specification of the component
 - Software design
 - Data model
- Doer: Development team

3. Test Levels

Unit Test/Component Test

- Component test may include:
 - Testing of functionality
 - Testing of specific non-functional characteristics, such as resource-behaviour (e.g. memory leaks) or robustness testing
 - Structural testing (Structure-based techniques)



3. Test Levels

Integration Test

- Integration test:

Test interfaces between components, interactions to different parts of a system such as an operating system, file system and hardware or interfaces between systems.

- Component integration test:

Tests the interactions between software components and is done after component testing.

- System integration test:

Tests the interactions between different systems and may be done after system testing.

- Doer: independent Test team

3. Test Levels

Integration Test

- Integration test may include:
 - Testing of functionality
 - Testing of specific non-functional characteristics (e.g. performance)
 - Structural testing (Structure-based techniques)
- IT is only after UT/CT



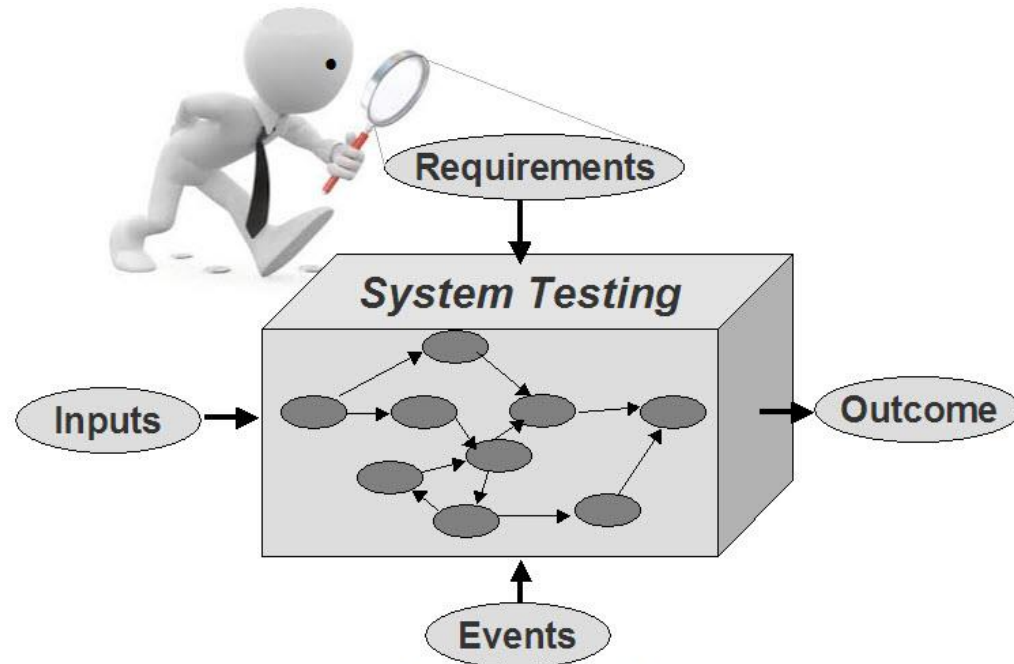
3. Test Levels

System Test

- System test:

The process of testing an integrated system to verify that it meets specified requirements.

- Often using the specification-based (black-box) techniques.



www.softwaretestinggenius.com

3. Test Levels

System Test

- System test:
 - Test environment should correspond to the final target or production environment as much as possible.
 - Should investigate both functional and non-functional requirements of the system
 - Independent test team often carries out system testing.
 - Structure-based techniques (white-box) sometimes may be used
- Doer: independent Test team

3. Test Levels

Acceptance Test

- Acceptance test:
Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.
 - Is often the responsibility of the customers or users of a system.
 - The goal is to establish confidence.
 - Finding defects is not the main focus in acceptance test.
- Doer: Customer/independent Test team



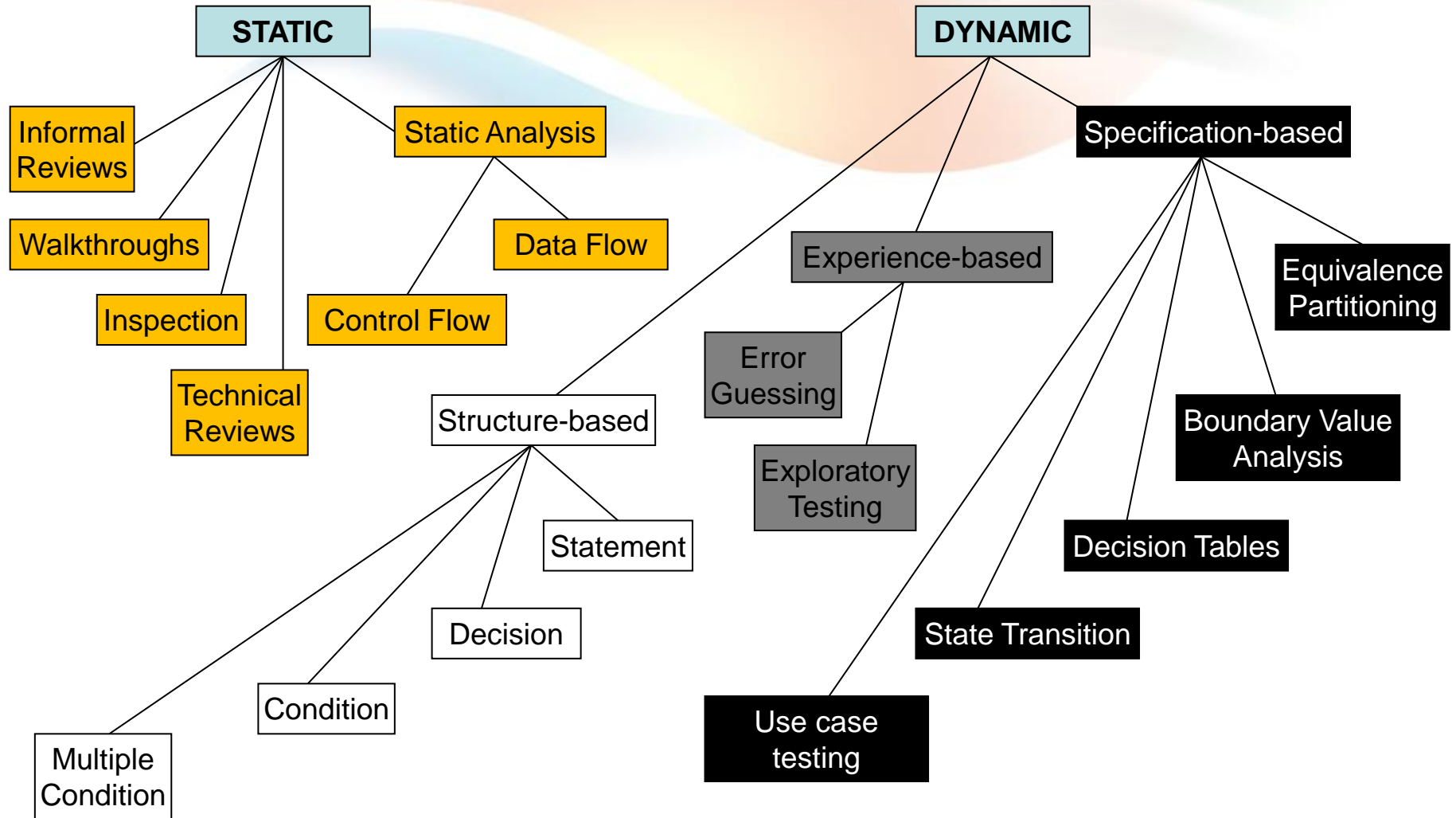
4. Test Types

- **Functional test:** testing of “what” the system does
 - Black-box testing
 - User Interface testing
 - Data and database integrity testing
 - Business cycle testing
 - Security testing
 - Interoperability testing
 - ...
- **Non-Functional test:** testing of “how” the system works
 - Performance testing
 - Usability testing
 - Maintainability testing

4. Test Types

- Non-Functional test: testing of “how” the system works (cont)
 - Load testing
 - Stress testing
 - Reliability testing
 - Portability testing
- Structural testing: white-box testing
 - What is happening 'inside the box'
- Testing relates to changes:
 - Re-testing: confirmation testing
 - Regression testing

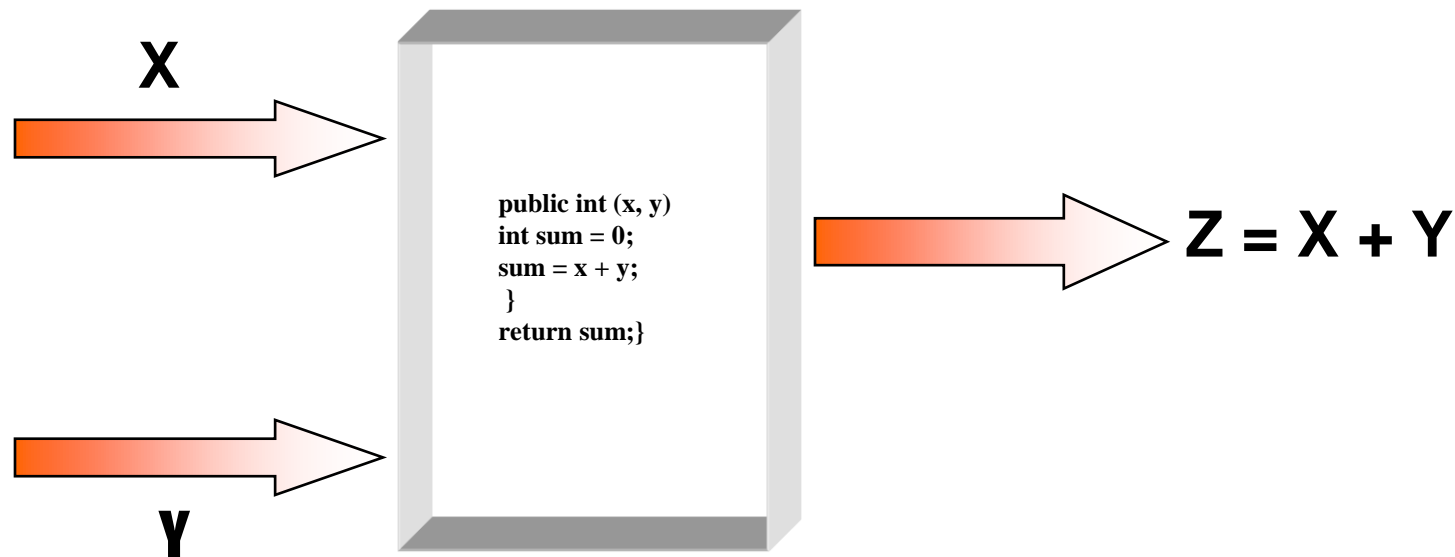
5. Test Techniques



5. Test Techniques

White Box Testing

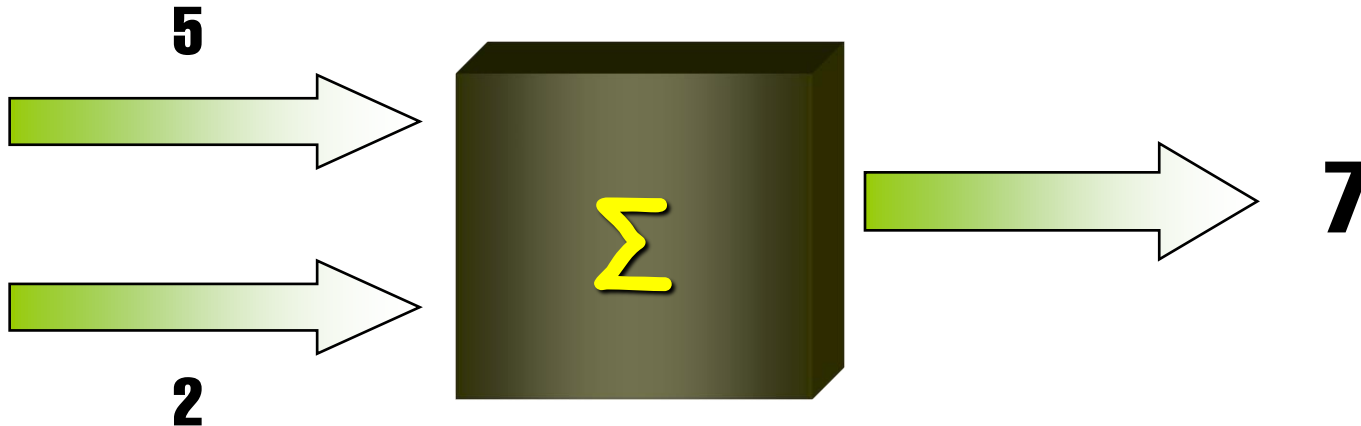
- White Box Testing:
 - Structural or structure-based techniques
 - Base on an analysis of the structure of the component or system.



5. Test Techniques

Black Box Testing

- Black Box Testing:
 - Specification-based techniques
 - Base on an analysis of the test basis documentation without reference to its internal structure.



5. Test Techniques

Experience Testing

- Experience Testing:
 - Tests are derived from the **tester's skill** and **intuition** and their **experience** with similar applications and technologies
 - Effectiveness of this technique depended on the testers' experience



6. Test Documents

Test Plan

- Test Plan to define:
 - Scope of test: levels and types of test
 - Risks may affect testing: quantify the loss of the risk
 - Training needs for testers
 - Requirements to test: what will be tested
 - Testing strategy: how will the test be performed
 - Completion criteria (Pass/Fail)
 - Tools to be used for testing
 - Testing environment and testing resource assignment (Responsibilities of testers)
 - Test Milestone: schedule of testing activities

6. Test Documents

Test Plan

- Test Plan to define:
 - Test deliverables: outputs of testing
 - Decide test metrics & manage through metrics
 - Criteria to stop testing:
 - Not good Unit Test
 - Low quality of code,...
 - Meet requested Test coverage
 - All Test cases are tested and passed



6. Test Documents

Test Design

- Specifications:
 - Basic flow
 - Alternative flow
- Test Design:
 - All test cases should be traceable to requirements
 - Design framework and rules for test cases (large/average items)
 - Plan number of test cases for normal, abnormal & boundary data
 - Assign testers for module/function/large items

6. Test Documents

Test Design

- Structure/Framework for Test cases (TC):
 - Modules or functions
 - Large items
 - GUI
 - Functional cases
 - Non-functional cases
- Number of TC:
 - Total
 - For each module
 - For each type: GUI/Function
 - For normal/abnormal/boundary cases

6. Test Documents

Test Case

- What is a test case?

A test case consists of a set of input values, execution preconditions, expected results and execution post-conditions, developed to cover certain test condition(s).

- Test case Sample:

TC ID	Test case Description	Test Data	Expected Result	Actual Result	Status	Priority	Comment
1. Verify New Account Setup function							
Precondition	- User has logged in to system - User is on New Account Setup screen						
Test case Title	Verify New Account Setup function -> Add Primary Contact information Precondition: User is on Primary Contact information screen						
1.1	Check mandatory fields: - Don't input any value on screen, click Next button		System will display message " Please choose an account plan."				
1.2	Input valid data into all fields, click Next button		- Data will save to Accounts table successfully - System will expand full screen with other contact information				

- Test case management tools: Test Link, Mercury,...

6. Test Documents

Test Case

- How is a good test case:
 - Should be accurate and tests what it is intended to test.
 - No unnecessary steps should be included in it.
 - It should be reasonable and reusable.
 - It should be closed to requirements.
 - It should be compliant to regulations.
 - It should be independent i.e. You should be able to execute it in any order without any dependency on other test cases.
 - It should be simple and clear, any tester should be able to understand it by reading once.
 - Cover all of cases that can occur

7. Defect Concepts

- What is defect?

A defect is any error found by testing and reviewing activities (all errors found by internal reviewer, external reviewer and customer).

- Where defects come from?

- Products

- ✓ SRS, DD, Source code, Test cases, etc.

- Quality Control

- ✓ Review, Test, Audit, Inspection

- Processes

- ✓ Requirements, Design, Coding, Test, etc.

- Other sources:

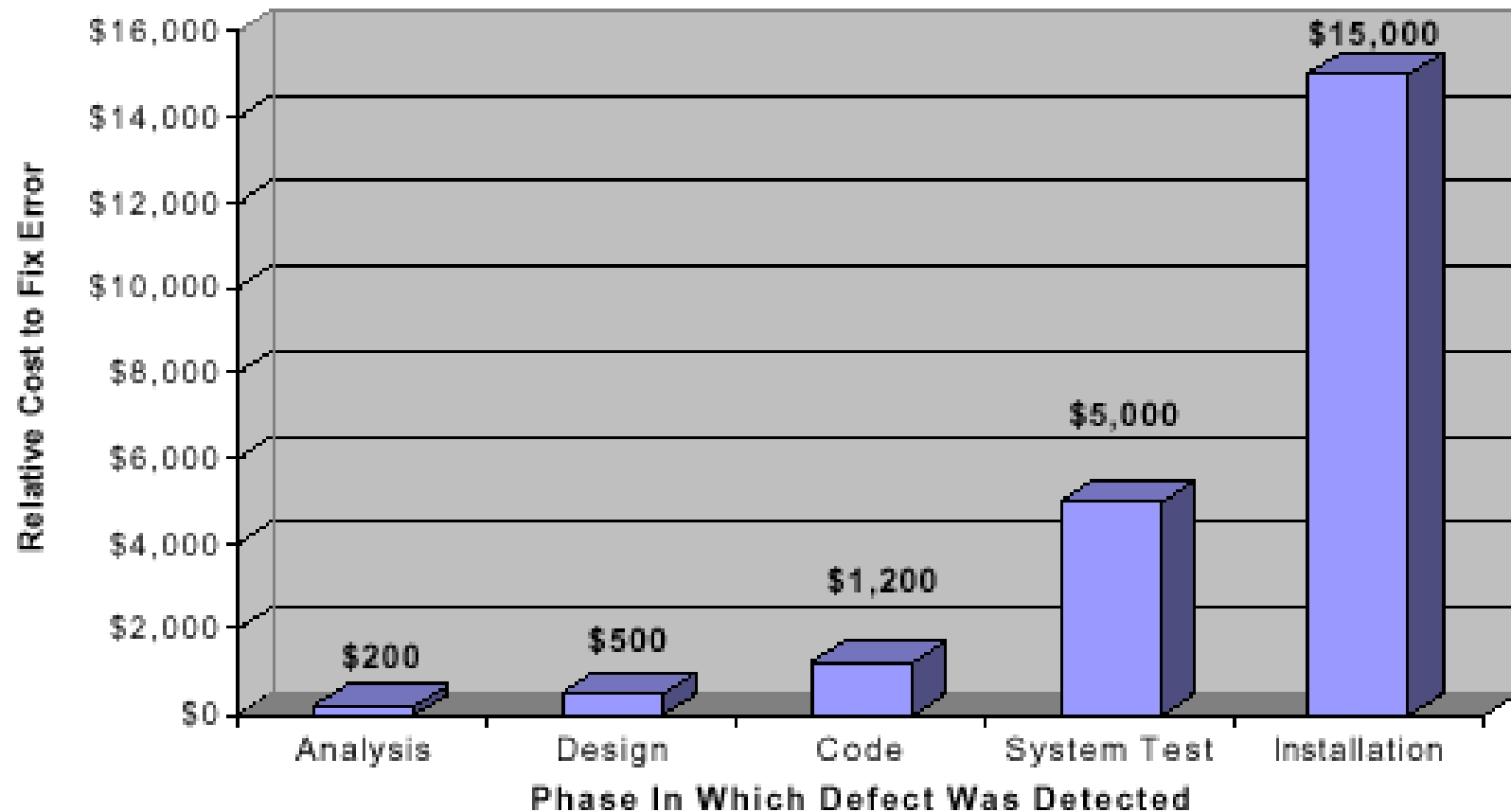
- ✓ Change requests, misunderstanding, carelessness, planning, etc.



7. Defect Concepts

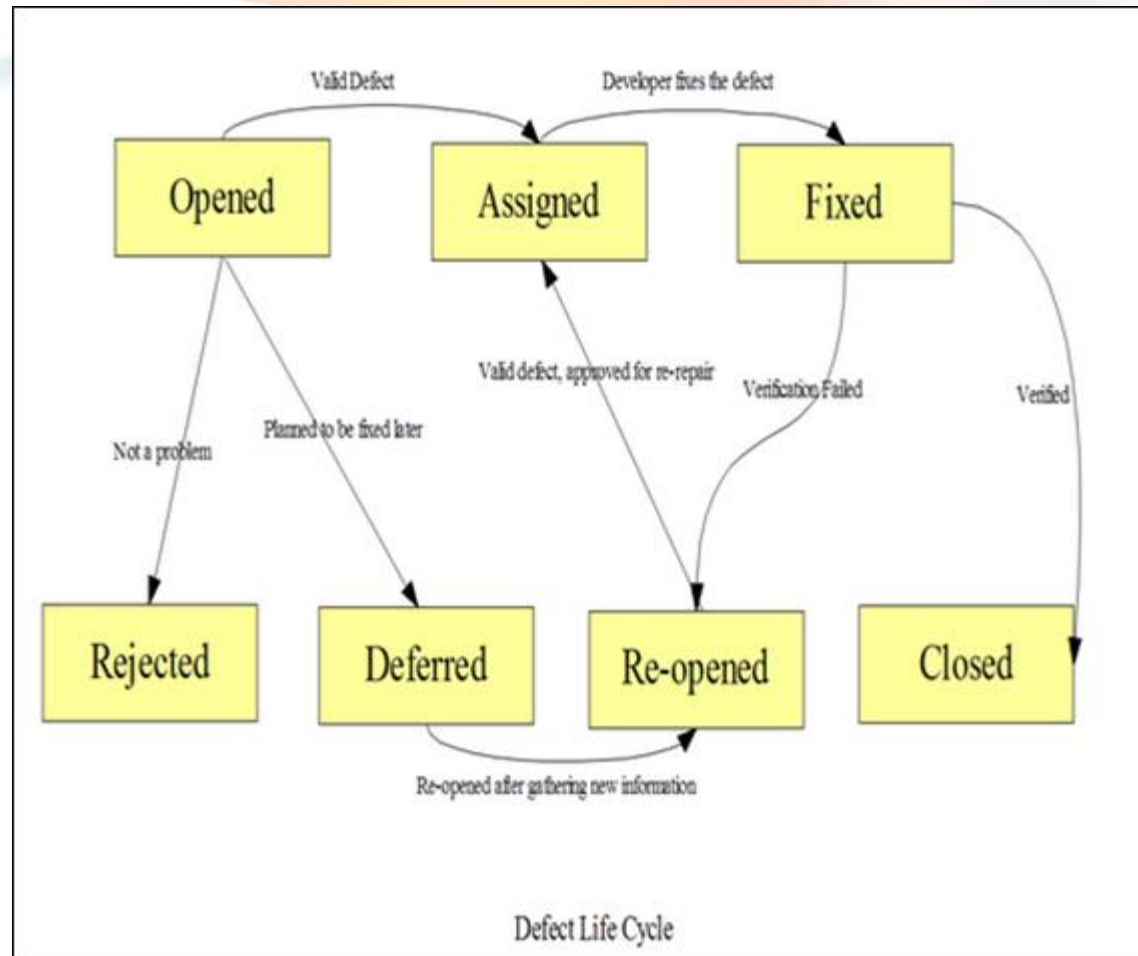
Cost of defects

Relative Quality Cost vs. Lifecycle Phase



7. Defect Concepts

Common Defect Life Cycle



7. Defect Concepts

Defect Content

- Main items in a defect:
 - Defect title
 - Component/Function
 - Defect description
 - Step to reproduce
 - Actual result
 - Expected result
 - Screen shot if any
 - Priority
 - Severity
 - Test case ID
 - Defect type

7. Defect Concepts

Defect Content

- Defect status:
 - Opened
 - Assigned
 - Fixed
 - Deferred
 - Re-opened
 - Closed
 - Reject (works as per Requirements, wont fix, works for me)

7. Defect Concepts

- How to write a good defect report:
 - General outline of a bug report:
 1. Summary/Title
 2. Component/Function Name
 3. OS (*optional*)
 4. Description
 5. Steps to Reproduce
 6. Actual Results
 7. Expected Results
 8. Attachments OS (*optional*)
- Defect summary: **Problem + Action + Location**
- Defect management tools: Bugzilla, Mercury,...

8. Test Measurements

Common Test Measurements

- Measurements:
 - Defects: number, severity, type of defects
 - Test effort: time spend for testing activities
 - Test coverage: number of executed test cases/ Total number of TCs
 - Test successful coverage: number of test cases executed pass/total number of test cases

9. Tester Attitudes and Skills

- **Attitudes:**

1. Careful
2. Patient
3. In details
4. Ethic

- **Skills:**

1. Communication
2. Reading Comprehension
3. Problem Solving
4. Documentation
5. Self-organization



Recommendation for Testers



1. Improve your foreign language skills
2. Self study software testing techniques
3. Improve the five skills
4. Attend software testing class
5. Take testing certificates



Thank you & Question - Answer

Reference:

1. <http://istqbexamcertification.com/>
2. <http://istqb.org/> (Syllabus Foundation Level)
3. <http://www.qatutorial.com>
4. <http://www.softwaretestingmentor.com>