



**talenttech**

# Software Testing Fundamentals

# Course Objectives

- ▶ Software testing concepts
  - Software Quality & Testing
  - Levels of Software testing
  - Types of Software Testing
  - Verification and Validation
  - Understanding of major activities
- ▶ Software Environments
- ▶ Software Test Managements
- ▶ Software Configuration Management
- ▶ A software testers job responsibilities



# Software Quality

## ▶ What is Software Quality?

- Quality software is reasonably bug-free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.
- Quality is obviously a subjective term. It will depend on who the 'customer' is and their overall influence in the scheme of things.



# Quality Assurance VS Testing

## ► Software Quality Assurance

- Involves the entire software development PROCESS – monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to '*prevention*'.

## ► Software Testing

- Involves operation of a system or application under controlled conditions and evaluating the results . The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they shouldn't or things don't happen when they should. It is oriented to '*detection*'.

# Verification & Validation



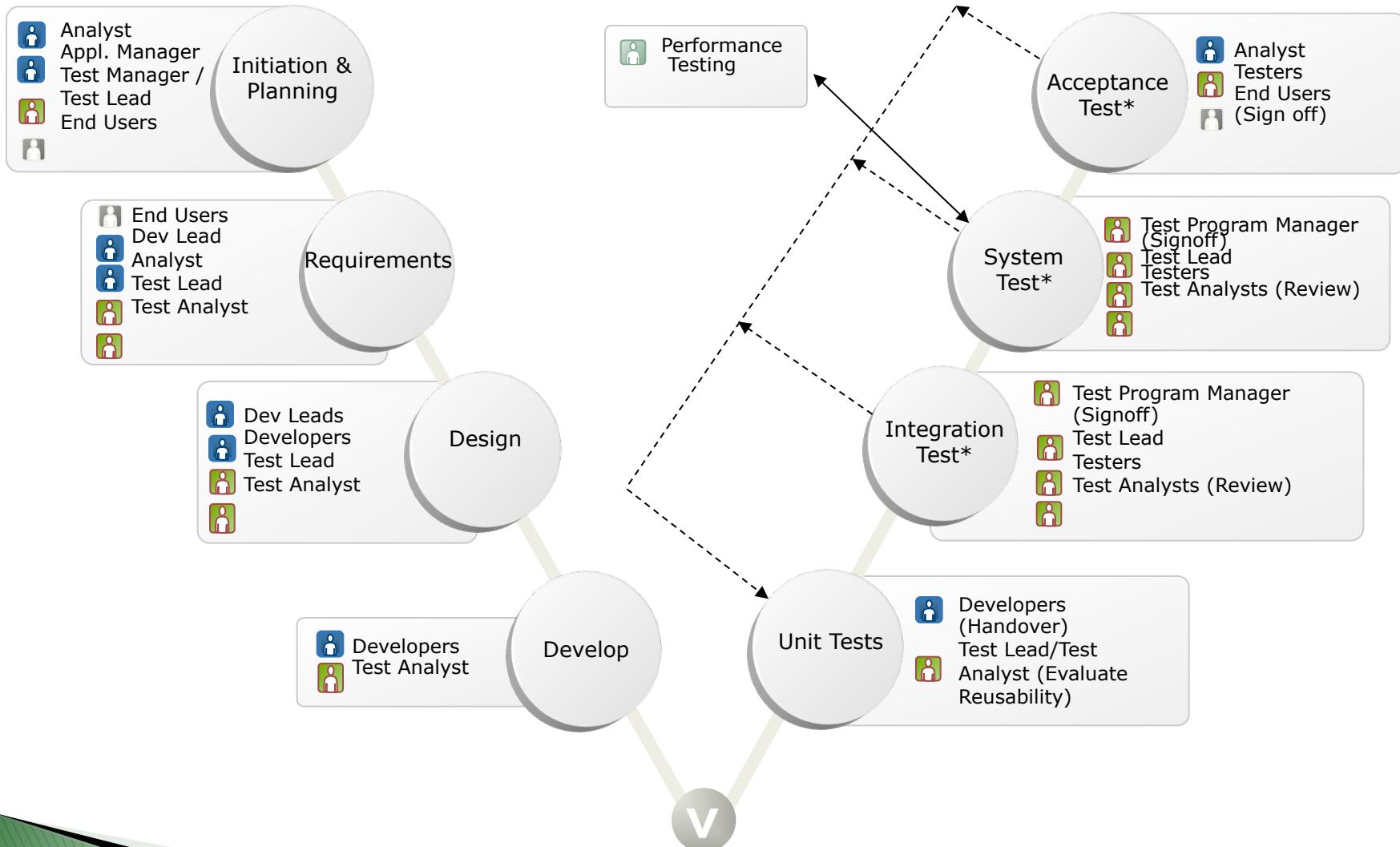
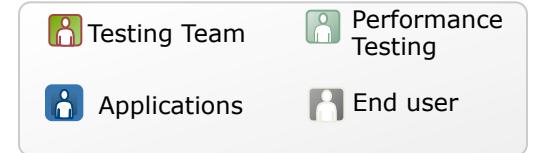
## ▶ Verification

- The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase (*We built the thing right*)

## ▶ Validation

- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements (*We built the right thing*)

# V - Model



\* Regression & Functional testing to be conducted as part of Integration, System & Acceptance test phases

# What is Software Testing?

- ▶ Software testing is the process of planning, preparing, executing, and analyzing aimed at establishing the characteristics of an information system, and demonstrating the difference between the actual status and the required status.

# Why Software Testing?

## ▶ Demonstration

- Gain confidence that systems can be used with acceptable risk.
- Try out features and functions under unusual conditions and situations.

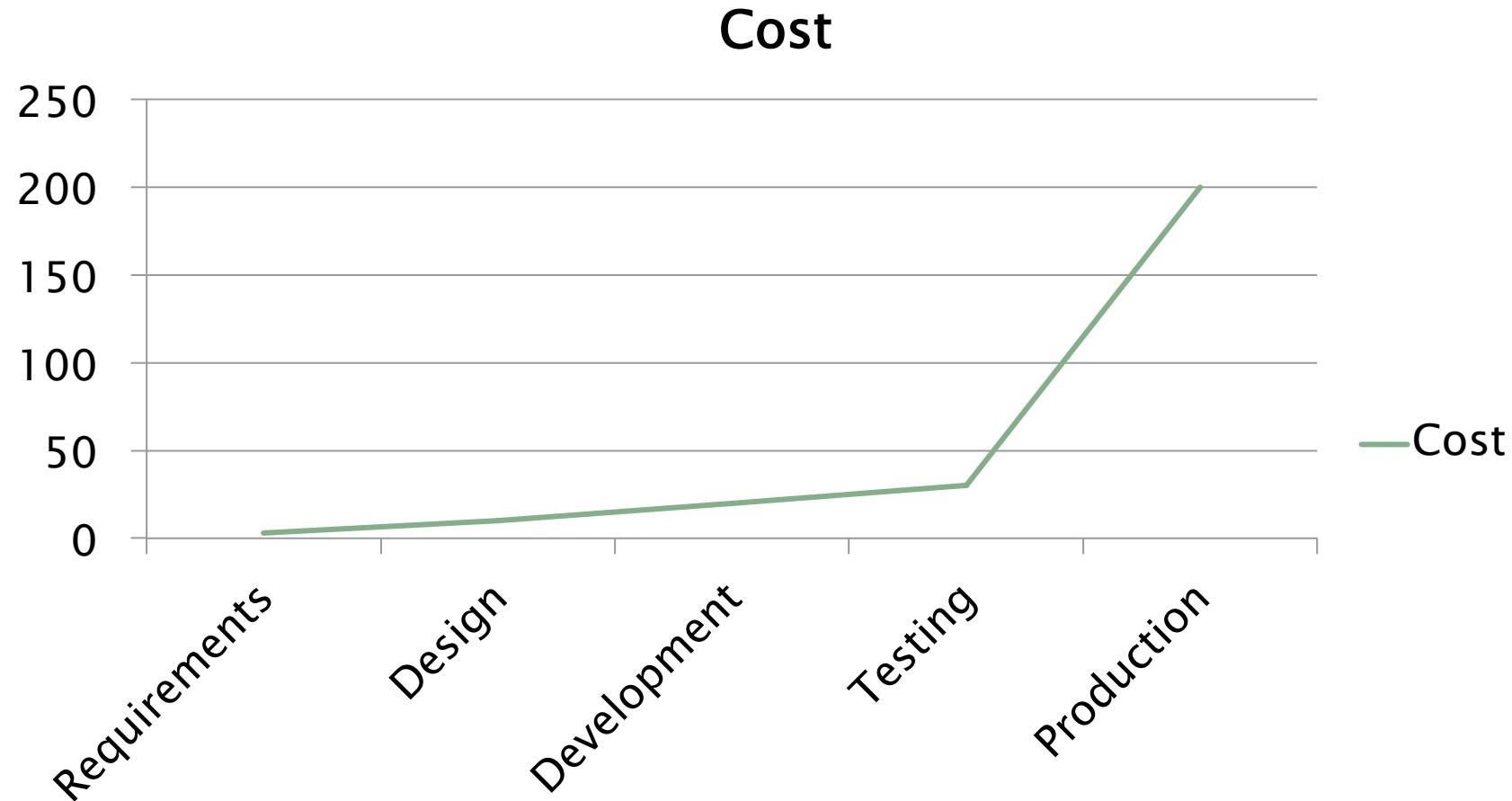
## ▶ Detection:

- Discover defects, errors, and system deficiencies.
- Define system capabilities and limitations.

## ▶ Prevention:

- Detect errors earlier in the process.
- Identify risks and problems and ways to avoid them in the future.

# The Cost of Software Errors





# Lesson

An effective software test team finds defect in early stages of software development lifecycle.

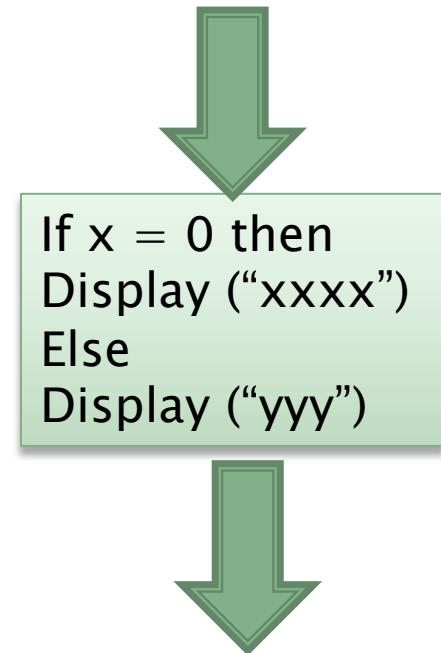
# Testing Methods – How?

- ▶ White box
- ▶ Black box
- ▶ Gray box



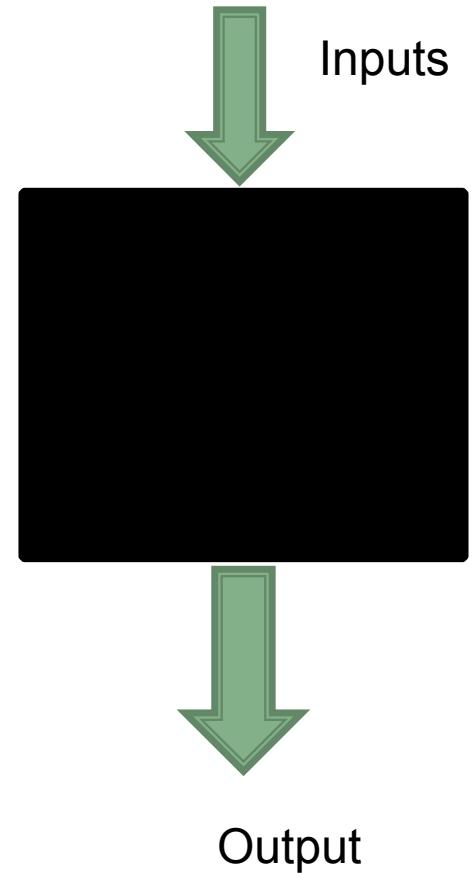
# White box/Structural testing

- ▶ A form of testing where a specific area within the code (Unit) is targeted for testing.
- ▶ Testing based on an analysis of internal workings and structure of a piece of software.
- ▶ A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data.



# Black box/ Behavioral testing

- ▶ Testing based on an analysis of the specification of a piece of software without reference to its internal workings.
- ▶ The goal is to test how well the component satisfy to the published requirements for the component.
- ▶ An approach to testing used by analyst and users alike where inputs and outputs of functions are known.
- ▶ Internal code structure is irrelevant



# Gray Box Testing

- ▶ Gray-box testing is a combination of White Box and Black Box Testing.
- ▶ The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications
- ▶ A gray-box tester partially knows the internal structure, which includes the access to internal structures as well as the algorithms for defining the test cases



# Remember

TalenTech focuses only on Black box Testing.

# Software Test Lifecycle(STLC)

## Requirements Analysis

- Good Requirements must meet three criteria: Clear, Consistency, and Testable

## Develop Plans

- Develop Master Test Plan following Standards
- Develop Test Cases, & Manage them using Tools such as QC
- Develop Requirements Coverage

## Execute Test Plans

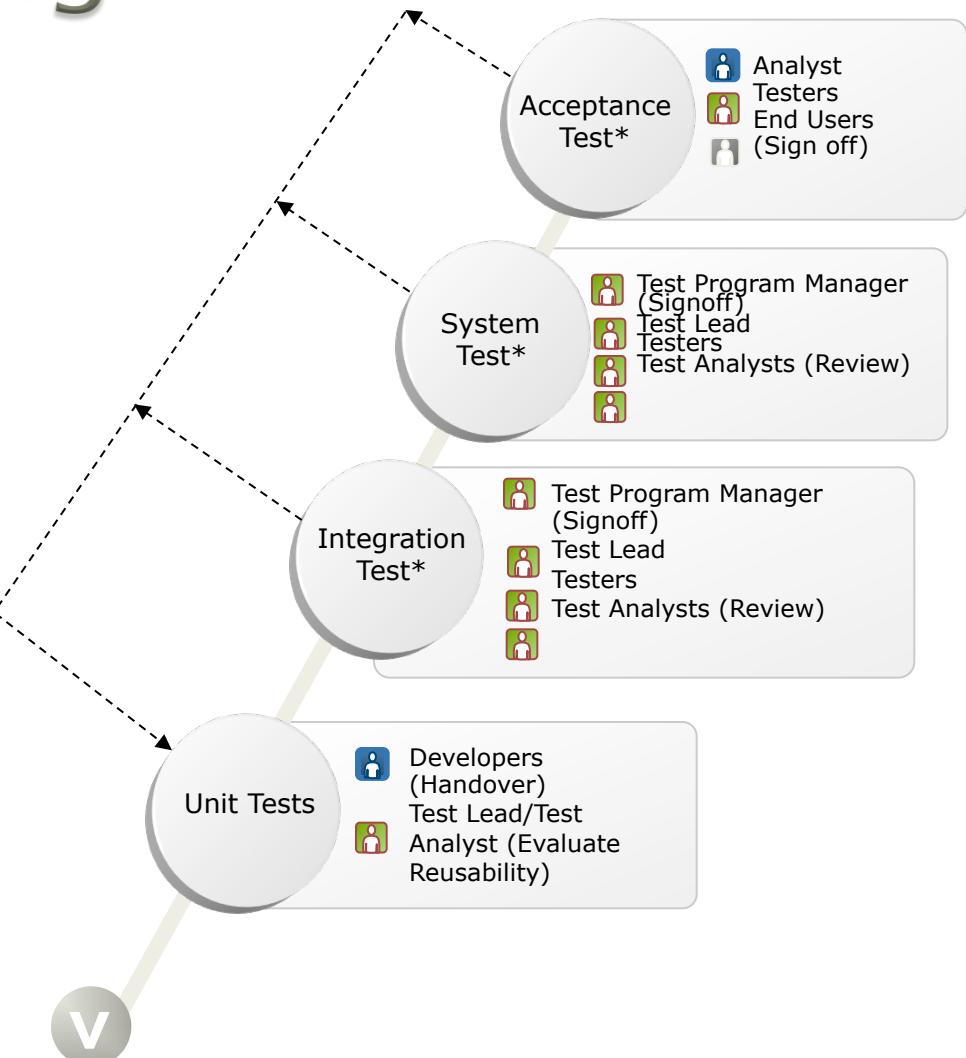
- Execute Test Cases which could be Manual or Automated Fashion
- Log Defects if Actual & Expected does not match
- Generate Execution Reports

## Manage Defects

- Establish Defects Life Cycle, and Review Board
- Re-Test Defects (Regression Test)
- Publish Reports

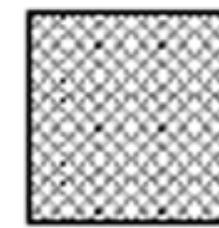
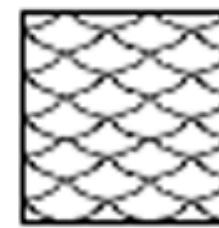
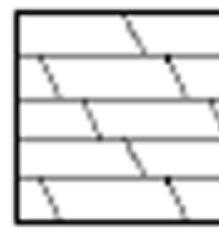
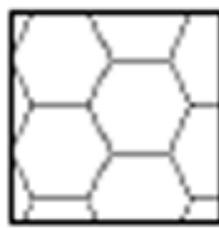
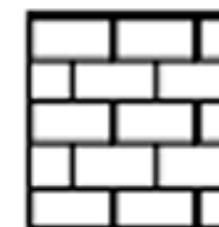
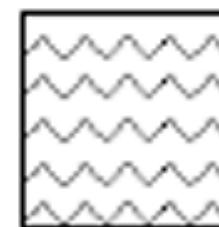
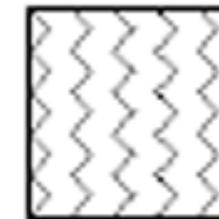
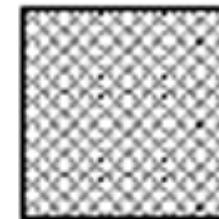
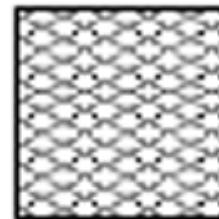
# Levels of Testing

- ▶ Unit Testing
- ▶ Integration Testing
- ▶ System Testing
- ▶ User Acceptance



# Unit testing

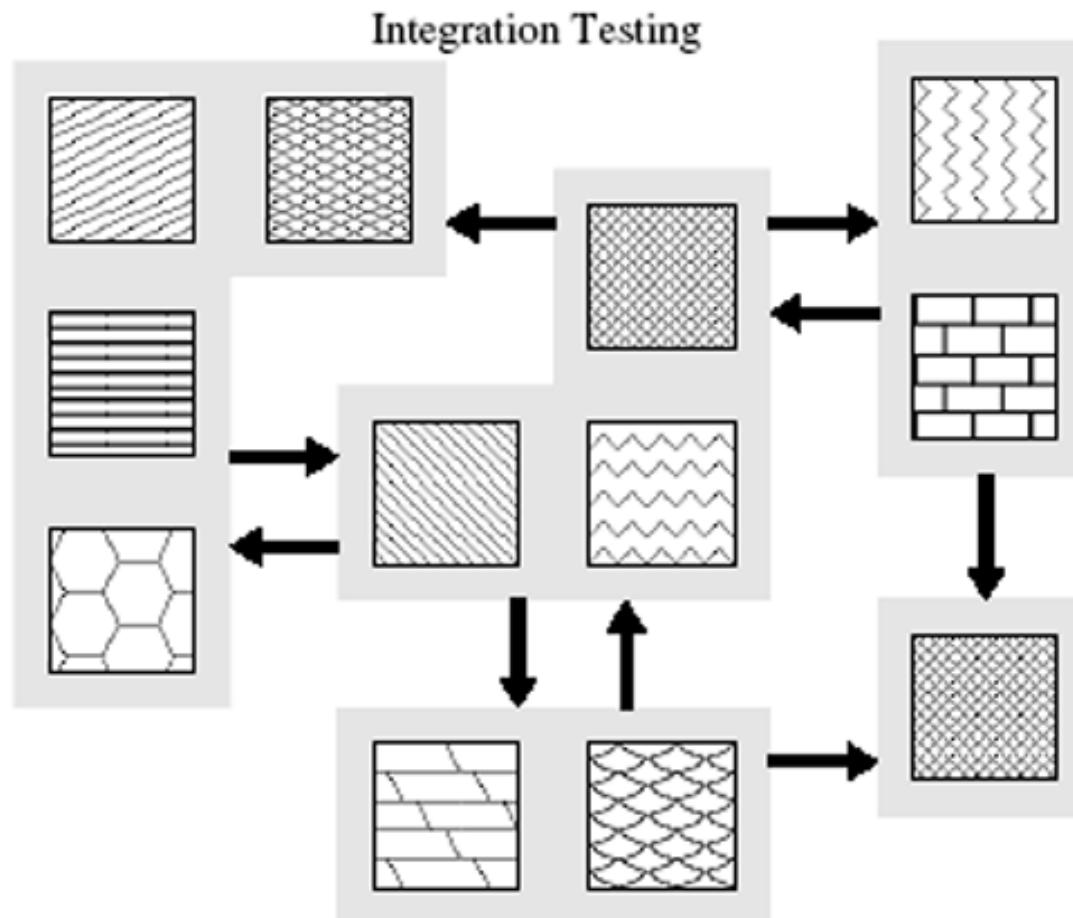
Unit Testing



# Unit Testing

- ▶ This is a white box testing technique where the developer has to look into the code and find out which unit/statement/chunk of the code is malfunctioning
- ▶ Unit Testing requires knowledge of coding and logic i.e. internal working of the code.
- ▶ Some projects may use the term Unit testing for black box while validating single requirements

# Integration Testing

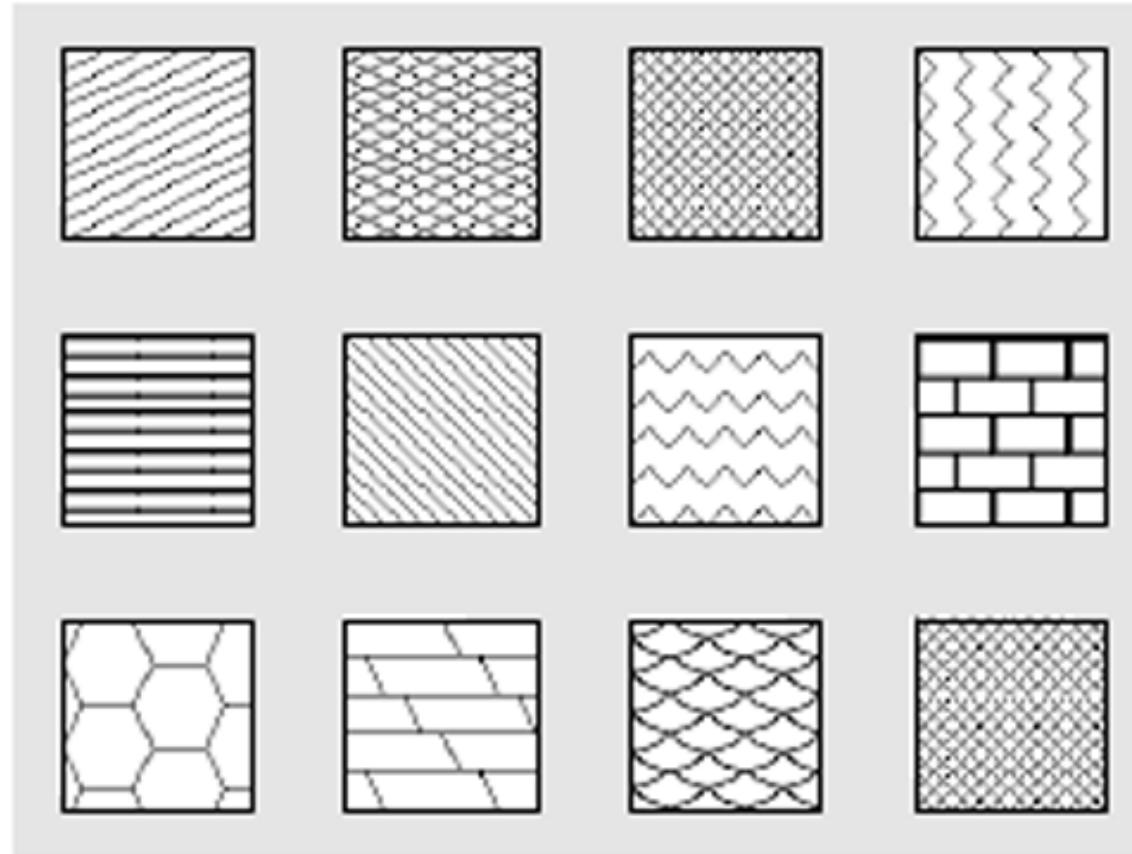


# Integration Testing

- ▶ Integration testing identifies problems that occur when units are combined. To ensure the viability of each before combining units, Testers know that any errors discovered when combining units are likely related to the interface between units.
- ▶ Testing in which software components are combined and tested progressively until the entire system has been integrated.

# System Testing

System Testing



# System Testing

- ▶ System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).
- ▶ System testing tests not only the design, but also the behavior and even the believed expectations of the customer.
- ▶ It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s)

# User Acceptance Testing (UAT)

- ▶ User Acceptance Testing (UAT) is a process to obtain confirmation that a system meets mutually agreed-upon requirements.
- ▶ A Subject Matter Expert (SME), preferably the owner or client of the object under test, provides such confirmation after trial or review.
- ▶ In software development, UAT is one of the final stages of a project and often occurs before a client or customer accepts the new system.

# UAT – How to Test?

- ▶ Test-designers draw up formal tests and devise a range of severity levels.
- ▶ Ideally the designer of the user acceptance tests should not be the creator of the formal integration and system test cases for the same system.
- ▶ The UAT emulates real-world usage conditions rather than negative scenarios
- ▶ Do not normally focus on identifying simple problems such as spelling errors and cosmetic problems, nor showstopper defects, such as software crashes

# Role of a Tester – UAT

- ▶ User Acceptance Tests (UAT) are usually performed by clients or end-users
- ▶ A senior Tester or UAT test engineer's job is to manage UAT testing activities such as providing the test cases, getting the environment ready, scheduling the sessions, monitoring the executions, and receiving the sign off

# Types of Testing

- ▶ Functional Testing
- ▶ Regression Testing
- ▶ Non-Functional Testing
- ▶ Smoke Testing
- ▶ 508 Testing
- ▶ Database Testing



# Functional Test

- ▶ Functional testing verifies that the system accepts the proper data and processes and retrieves the data based on the appropriate business rules.
- ▶ Function Testing will also include retesting of the defects identified in prior cycles, using similar test scripts and processes
- ▶ Functional testing focuses on the followings:
  - Positive
  - Negative, and
  - Boundary Scenarios

# Functional Test – Positive

- ▶ Positive testing validate the application as per requirements
- ▶ Positive test are conducted using valid data
- ▶ Example of positive tests:
  - Test username and password using valid values such as lengths and character allowed
  - Accepting a payment using a valid credit card number
  - Reserve a flight using valid dates

# Functional Test – Negative

- ▶ Negative testing validates the application against requirements
- ▶ We perform negative testing using invalid data
- ▶ Example of negative testing
  - Test using expired or invalid credit cards
  - Enter a wrong password to login to system
  - Reserve a flight using past dates
  - Submit a form without entering any data or invalid data

# Functional Test – Boundary

- ▶ Boundary testing validate an application based on the
- ▶ A boundary is a place where the system's expected behavioral changes
- ▶ A boundary exists at the point at which an input value goes from valid to invalid
- ▶ Example of boundary values
  - 0, -1 for number of order
  - Yesterday for flight reservation
  - 99 for a two digit values

# Regression Test

- ▶ Regression testing ensures that a change or a defect fix has not caused faults to appear in unchanged parts of the system.
- ▶ Regression is generally associated with some change to the system, such as adding a feature or fixing a bug.
- ▶ Regression test cases will be developed in such a way that the test covers overall functional scenarios of the application.

# Regression – Fix

- ▶ Whenever defects are fixed by the developer, it is assigned back to the test analyst for retest to verify.
- ▶ Regression Fix confirms that a developer successfully fixed the defect.
- ▶ Test analyst who opened the defect will execute the same test case that identified the defect in the first place.
- ▶ After verifying, test analyst will close the defect.

# Regression – Risk

- ▶ Some defects may hide within defects or cause a defect because of another code fix or enhancements
- ▶ During defect fixes the development team may unintentionally introduce new errors into the code.
- ▶ Regression risks make sure that the impacted areas of the application are tested properly because of an defect fix
- ▶ The Test team conduct an impact and risk analysis for the defects and identify additional test cases/ scripts that should be run based on code complexity, defect density, and code priority.
- ▶ The impact analysis provides the number of test scripts and the level of effort to run the test cases for the current build as well as for the previous builds.

# Regression Strategy

- ▶ Repeat All Tests – Many projects use the strategy to execute all tests after every new build to ensure that all regression defects are found. This is time consuming but an effective approach using test automation such as QTP.
- ▶ Repeat Some Tests – It is not always possible to repeat all tests even with test automation. Using traceability and change analysis, some project repeat a select number of test cases.

# Non Functional Test

- ▶ Non functional testing tests quality characteristics, such as reliability or usability rather than the features of the application.
- ▶ Different Types of Non Functional testing are:
  - ▶ Performance/ Load Testing
  - ▶ Usability Testing
  - ▶ Security Testing
- ▶ TalenTech offer performance/ load testing class using HP LoadRunner

# Smoke Test

- ▶ Smoke is the initial level of testing effort to determine if the new software version is performing well enough for its major level of testing effort.
- ▶ The Smoke test scenarios should emphasize breadth more than depth.
- ▶ All components should be touched, and every major feature should be tested briefly.
- ▶ If test fails, the build is returned to developers un-tested.
- ▶ The main purpose of Smoke testing is to find system crashes, and environmental failures.

# 508 Testing

- ▶ Section 508 Compliance Testing ensures that the software meets 508 Compliance requirements.
- ▶ Section 508 was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals.
- ▶ The law applies to all Federal agencies when they develop, procure, maintain, or use electronic and information technology.
- ▶ For more info Visit: <http://www.section508.gov>

# Database Testing

- ▶ Database Testing includes the testing of actual data (content) and database integrity
- ▶ We perform database testing to ensure that data is not corrupted and the schema are correct, as well as the functionality testing of the database applications
- ▶ SQL scripting is generally used to test databases
- ▶ TalenTech covers basic to advanced SQL using Oracle



# Testing Activities

Responsibilities of a tester.

# Major Activities in Testing

- ▶ Requirements Analysis
- ▶ Test Planning and Design
- ▶ Test Estimation
- ▶ Test Data Creation
- ▶ Develop Test Cases
- ▶ Requirements Tracing
- ▶ Test Execution
- ▶ Managing Defects
- ▶ Test Reporting



# Requirement Analysis

- ▶ A software test analyst/ engineer analyze requirements from a testers perspective that includes
  - Whether the requirements are clear
  - Whether the requirements are consistent, and
  - Requirements must be testable



# Test Planning – Master Test Plan

1. Introduction
2. Test Items
3. Software Risk Issues
4. Features to be Tested
5. Features not to be Tested
6. Approach
7. Item Pass/Fail Criteria
8. Suspension Criteria and Resumption Requirements
9. Test Deliverables
10. Remaining Test Tasks
11. Environmental Needs
12. Staffing and Training Needs
13. Responsibilities
14. Schedule
15. Planning Risks and Contingencies



# Sample Test Plan

Attached at the end of class materials or  
Visit [www.TalenTech.Org](http://www.TalenTech.Org)

# Test Estimations

- ▶ Test estimations includes the following factors
  - Requirement analysis
  - Test case/ data developments
  - Test execution
  - Test reporting, and
  - Defect tracking
- ▶ Test estimations are the number of hours to perform testing to validate a requirements

# Test Data Creation

- ▶ Test data are used to test requirements, and execute a test case
- ▶ There are different ways of creating test data
  - Scrambled production data
  - Developing data based on application
  - Data given by business analyst or customer
  - Specific data arranged for testing such as credit cards, bank accounts, driving license by environment support team or specific departments
- ▶ Test analyst/ engineers are responsible to create/ collect test data

# Test Case

- ▶ A test case is a document that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly.
- ▶ A test case should contain particulars such as
  - Test case identifier
  - Test case name
  - Objective
  - Test conditions/setup
  - Input data requirements/ Precondition
  - Action/ Description
  - Expected results
  - Comments

# When to Develop Test Cases

- ▶ You may develop a test case long before the application is ready
- ▶ Test cases are developed as soon the detail levels of Systems/ Functional requirements are ready and analyzed
- ▶ You may develop test cases based on systems/ functional requirements, or test cases and detail user interface design
- ▶ In Agile, a developer write codes based on test cases

# Sample Test Case

Requirement: User should be able to find a correct zip code based on address, city, and state

Step Name	Description	Expected Results	Actual Results
Step 1	Launch a browser, and enter <a href="http://www.usps.com">www.usps.com</a>	USPS home page appears	
Step 2	Click on "Look up a Zip code" link	Zipcode lookup page appears	
Step 3	Enter street, city, and click submit	Error message displays	
Step 4	Enter street, state, and click submit	Error message displays	
Step 5	Enter street, city, state, and click submit	Correct zip code displays	



# Assignment

1. Develop a test case based on the requirements provided. Use MS Excel.
2. Create two sets of test data for the test case execution. Use MS Excel.

# Requirement Traceability

- ▶ Requirement Traceability Matrix – trace requirements through software development life cycle stages such as design, development, and testing.
- ▶ RTM is developed by business analyst and maintained by the SDLC teams

# RTM Example

- ▶ RTM has following fields
  - Requirement ID
  - Description
  - Priority
  - Status – Proposed, Approved, Incorporated, Validated
  - Design
  - Test Case

Req ID	Description	Use Case	Design	Test Case
SRS 1	Test requirement 1	UC 1	RD 102	TC 1
SRS 2	Test requirement 2	UC 2	RD 103	TC 2

# Test Execution

- ▶ Test execution is the process of following the steps of a test case and document the actual results
- ▶ Executing a test case have the following outcomes
  - Pass: When actual results match expected results
  - Fail: When Actual and Expected results do not match. In this case you must open a defect.
  - Incomplete: When the results are inconclusive



# Assignment

1. Update RTM based on the Test Case Developed
2. Execute the developed Test case and document Expected and actual results.

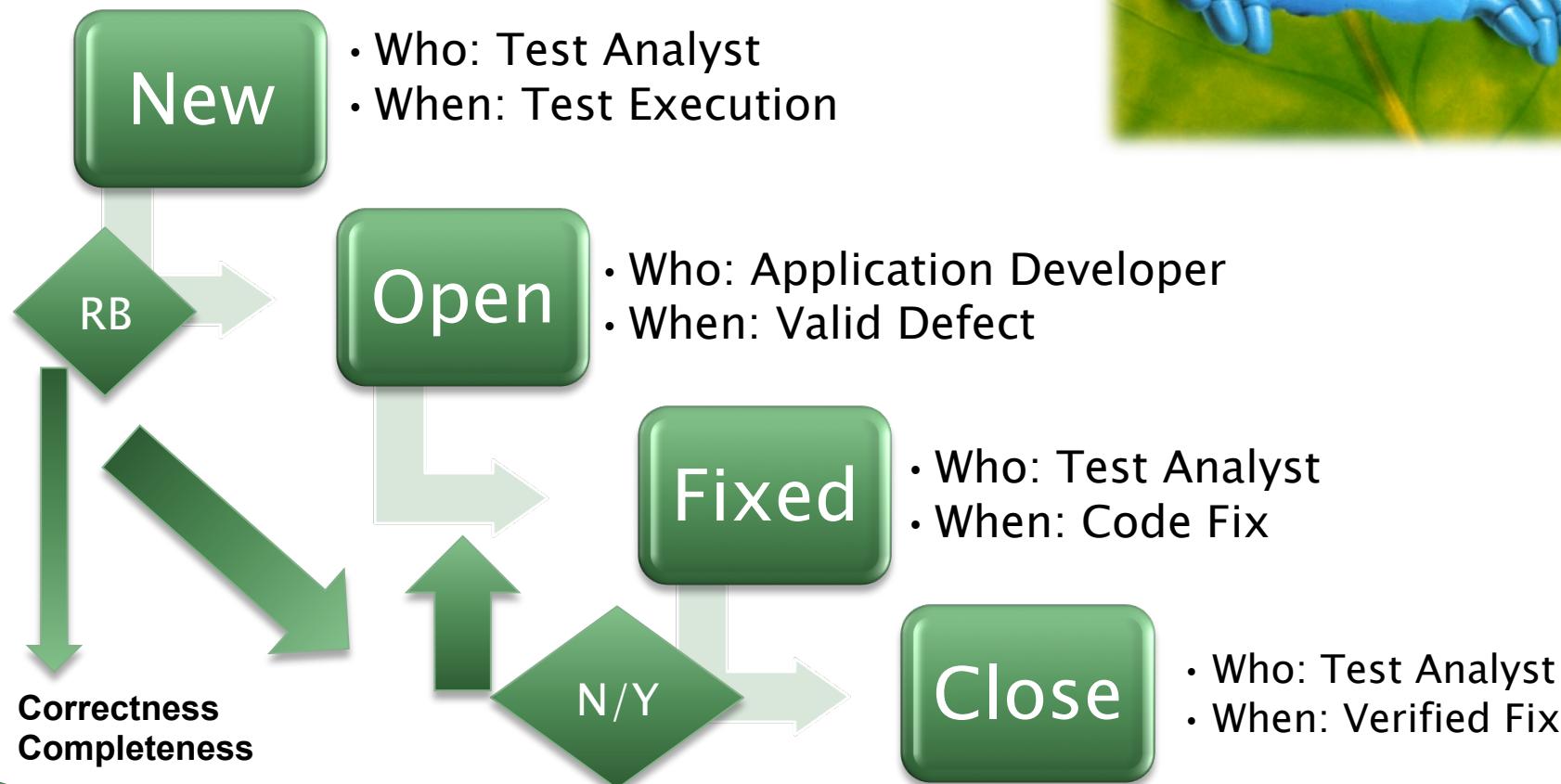
# Working with Defects

- ▶ A **software Defect/ bug** is the common term used to describe an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways
- ▶ Defect tracking is important as complex software systems typically have tens or hundreds or thousands of defects
- ▶ When the numbers of defects gets quite large, and the defects need to be tracked over extended periods of time, use of a defect tracking system can make the management task much easier.

# Defect Description

- ▶ A defect description must have the followings
  - **Subject**: A single sentence explaining what the defect is about
  - **Status**: Defect status based on the defined DLC
  - **Detected By**: The owner of the defect
  - **Version**: What is the current version of the app
  - **Severity**: How serious the defect is
  - **Reproducible**: If the defect can be reproduced
  - **Detection Date**: When the defect found
  - **Assignee**: Who will be assigned to work on next
  - **Description**: Detail of the defect that will include steps to reproduce

# Defect Management





# Assignment

Open a new defect based on the test case executed. Follow the proper defect description, and document it in MS Word.

# Test Reporting

- ▶ Test teams are responsible to report test activities and software quality matrix in a regular fashion
- ▶ The target audiences of the reports are managers, directors, CIO, customers, and the SDLC teams
- ▶ The reports may include the followings
  - # of TC executed, passed, failed, Not complete
  - # of Defects found with priorities and severities
  - Summary by Release, testing types, and period

# Test Managements

- ▶ Areas of software test managements are:
  - Requirements Managements
  - Test Plan Managements
  - Test Execution
  - Defect Managements, and
  - Real time reporting on software quality
- ▶ Test management is interrelated to configuration management, and software environments
- ▶ TalenTech offers classes on test management tools such as Quality Center, and Rational

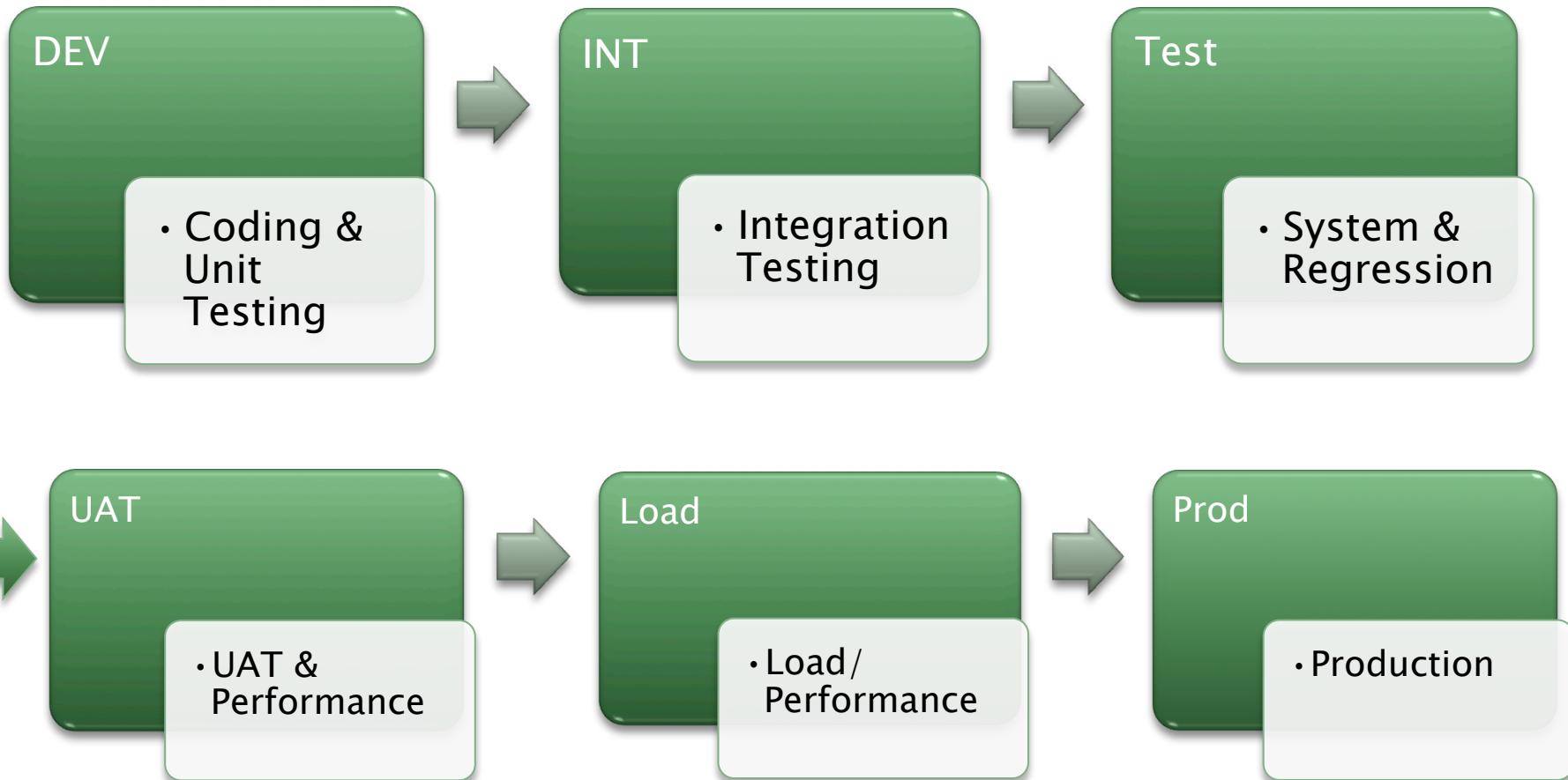
# Configuration Management

- ▶ Configuration Management (CM) is the art of identifying, organizing and controlling modifications to the software being built by SDLC team
- ▶ According to SEI: The purpose of CM is to establish and maintain the integrity of work products.
- ▶ CM identify and document all proposed changes and perform impact analysis
- ▶ CM performs auditing of developments and testing to ensure that the teams follow standards set by CM team

# Why do we need CM?

- ▶ The latest version of code cannot easily be found
- ▶ A difficult defect that was fixed at great expense suddenly reappears
- ▶ A fully tested feature is mysteriously does not work
- ▶ The wrong version of the code was tested
- ▶ There is no traceability between requirements, documentation, and code
- ▶ The wrong versions of the product being delivered

# Software Environments





# Now What?

To be or not to be...

# Testers Job Responsibilities

- ▶ A testers Job varies based on the types of testing.
- ▶ Common responsibilities includes writing test plan, test case, requirement traceability matrix, executing test cases, and submitting defects.
- ▶ A tester's job is not to prove the application works but to prove it doesn't work.
- ▶ Look for the key problems in the product with all your creativity and skill.



# Qualities of a Test Engineer

- ▶ A good test engineer has a 'test to break' attitude
- ▶ An ability to take the point of view of the customer, a strong desire for quality, and an attention to detail.
- ▶ Tact and diplomacy are useful in maintaining a cooperative relationship with other SDLC teams



# Qualities of a Tester – Cont'd

- ▶ Ability to communicate with both technical and non-technical (customers, management) people
- ▶ Analytical and Judgment skills are needed to assess high-risk areas of an application on which to focus testing efforts when time is limited.





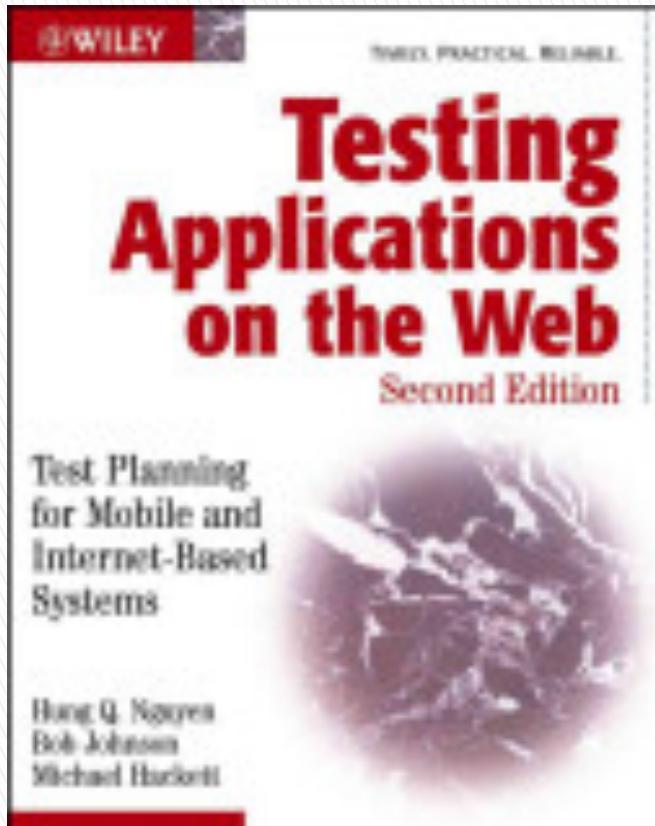
# Remember

Software Testing is NOT a hard core technical job, but ensuring the quality of a software that meets clients need.

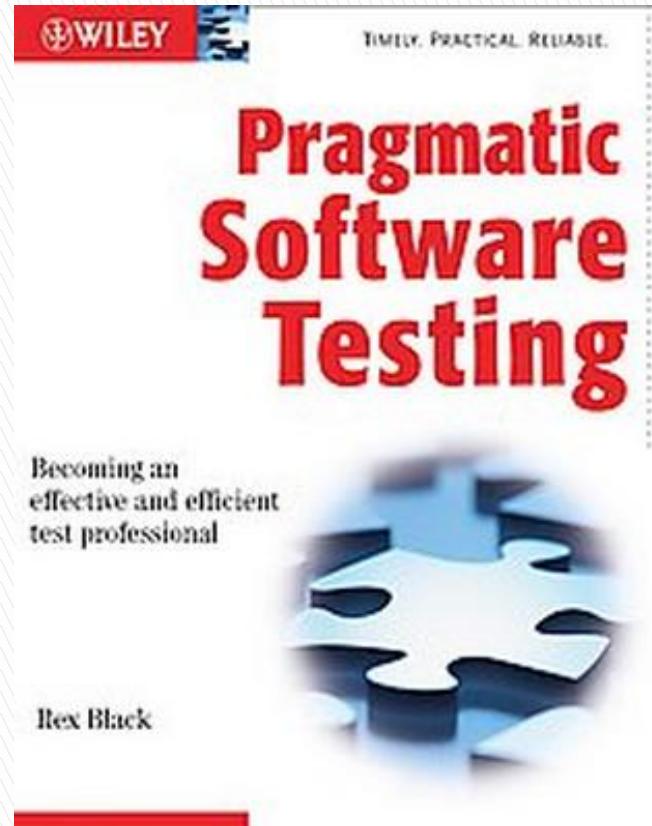
# Homework

- ▶ Study class notes
- ▶ Complete Homework Assigned
- ▶ Prepare 5–7 minutes presentation on the followings – Prepare for next class
  - SDLC Stages
  - Explain and compare methodologies
  - Explain V Model and Testers role on the stages
  - Levels of testing
  - Software Test Life Cycle
  - Defect Life Cycle
  - Requirement Coverage/ Traceability/ RTM

# Recommended Books



Testing Applications on  
the Web



Pragmatic Software  
Testing

# Interview Questions

Prepare before Next Class...

# Sample Interview Questions

1. What is Software Quality Assurance(QA) and Testing?
2. What are the software development stages?
3. Explain Software Test Life Cycle (STLC)?
4. What is a software requirement? What are different types? Give example.
5. What is RTM? What is your role in RTM?
6. What is a Test Case? What a test case must have?
7. Define smoke, functional, regression, System, and UAT testing.
8. Explain the defect life cycle.
9. What are the qualities of a good software tester?
10. What is scrum?

