Internship on Software Testing

Dr. Harshali P. Patil
Thakur College Of Engineering & Technology

Course Contents:

Module Number	Module Name	Detail contents	Hours
1	Basics of testing	Introduction to testing, types of testing , writing test cases for various module. Exercise to manually test the College website and report errors. Difference between manual and Automated testing, Automation testing tools, use of automation testing	10
2	Introduction to Selenium	Selenium Suite tools, Install Selenium IDE	5
3	Selenium IDE	Basics of Selenium IDE, test with Selenium IDE	5
4	Selenium Commands – Selenese	Selenium Commands – Selenese(Actions, Assessors and Assertions) , Create a script manually and test it on any website	15
5	Selenium Webdriver	Selenium Webdriver, Web driver commands and its implementation., Writing script using a webdriver, Use of findElement() and findElements() methods in Webdriver	15
6	Case Study	Write a complete test script for testing of a website	10
Total Hours 60 Hours			

Module 1: Basics of Testing Content

- Introduction
- Objectives
- Ways of Software Testing
- Testing Life Cycle
- Testing Methodologies
- Types of Testing
- Levels of Testing
- Tools for Automated Testing

INTRODUCTION

- It is the process used to identify the correctness, completeness and quality of developed computer software.

- It is the process of executing a program/application under positive and negative conditions by manual or automated means. It checks for the :-

- Specification
- Functionality
- Performance

OBJECTIVES

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

Error, Bug, Fault & Failure

Error : It is a human action that produces the incorrect result that produces a fault.

Bug : The presence of error at the time of execution of the software.

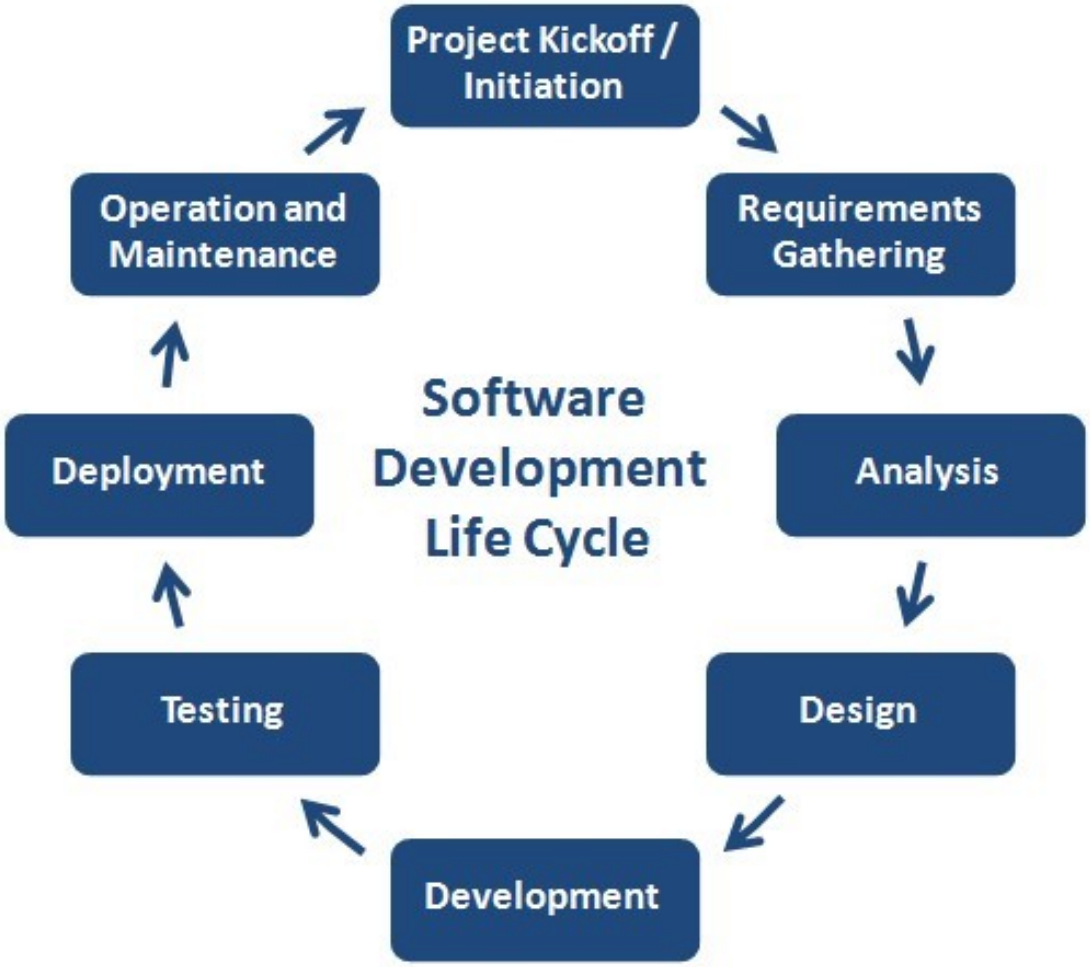
Fault : State of software caused by an error.

Failure : Deviation of the software from its expected result. It is an event.

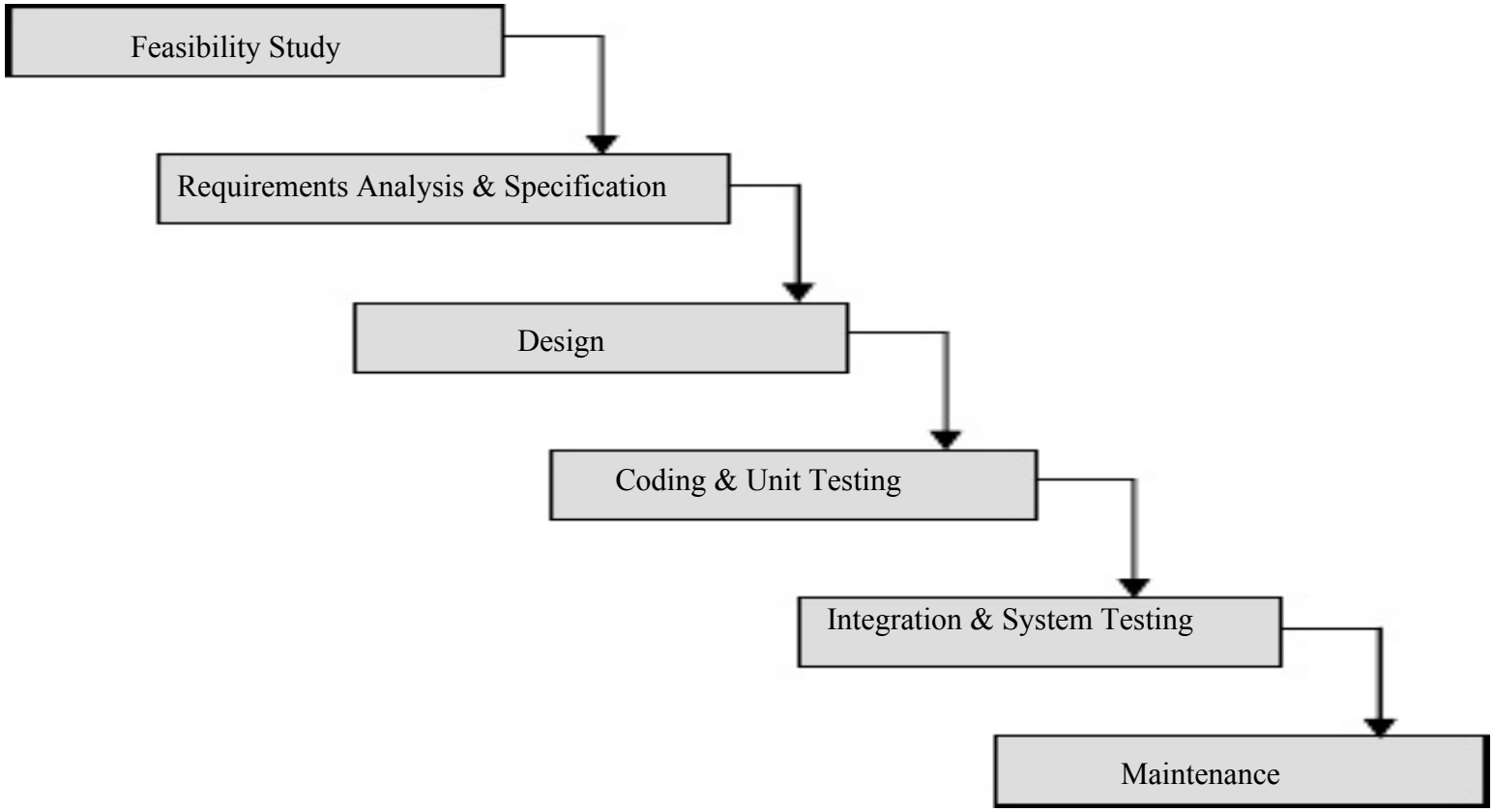
SDLC(Software Development Life Cycle)

- Standard model used world wide to develop a software.
- A framework that describes the activities performed at each stage of a software development project.
- Necessary to ensure the quality of the software.
- Logical steps taken to develop a software product.

SDLC(Software Development Life Cycle)



Classical Waterfall Model



It is the oldest and most widely used model in the field of software development.

What is Software Testing?

Software Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is **defect free**.

Why is software testing necessary?

1. To gain customer confidence
2. To check software adaptability
3. To identify errors
4. To avoid extra costs
5. To accelerate software development
6. To avoid risks
7. To optimize business

Who does Testing?

- Software Testers
- Project Developers
- Project Lead
- Team Manager
- End Users

Ways of Testing

Manual Testing

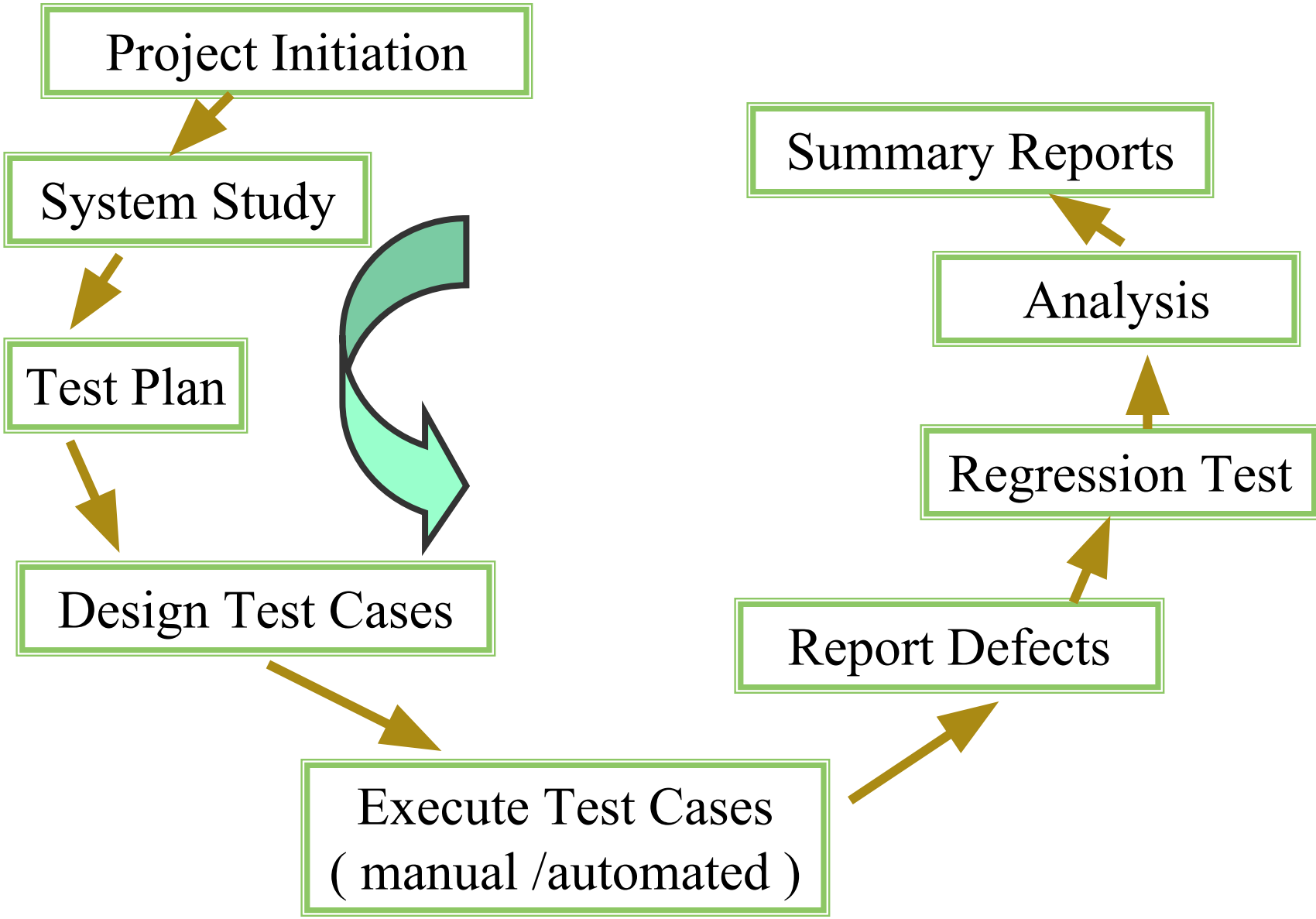
VS

Automated Testing

Manual Testing vs. Automation Testing

Manual Testing	Automated Testing
1.Manual testing requires human intervention for test execution.	Automation Testing is use of tools to execute test cases
2. Manual Testing is prone to human errors. Therefore the accuracy and reliability of the test results are low.	Automated Testing, on the other hand, is more reliable. Tools and/or scripts perform the automated test.
3.The cost of manual testing is dependent upon the human resources deployed in testing.	The cost of automated testing is dependent upon the testing tools deployed for performing tests.
4. The time required for manual testing is more as human resources perform it.	The time required for automated testing is less as software tools execute the tests.
5. Manual testing is suitable when the test cases are run once or twice. Therefore there is no frequent repetition of test cases.	Automated testing is suitable when the test cases need to run repeatedly for a long duration of time. Automated testing is recommended only for stable systems and is mostly used for Regression Testing.

Testing Life Cycle



Verification vs. Validation

- **Verification**: The software should confirm to its specification (Are we building the product right?)
- **Validation**: The software should do what the user really requires (Are we building the right product?)

Verification vs. Validation

Verification	Validation
1. Verification is a static practice of verifying documents, design, code and program.	1. Validation is a dynamic mechanism of validating and testing the actual product.
2. It does not involve executing the code.	2. It always involves executing the code.
3. It is human based checking of documents and files.	3. It is computer based execution of program.
4. Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	4. Validation uses methods like black box (functional) testing and white box (structural) testing etc.
5. **Verification** is to check whether the software conforms to specifications.	5. **Validation** is to check whether software meets the customer expectations and requirements.
6. It can catch errors that validation cannot catch. It is low level exercise.	6. It can catch errors that verification cannot catch. It is High Level Exercise.
7. Target is requirements specification, application and software architecture, high level, complete design, and database design etc.	7. Target is actual product-a unit, a module, a bent of integrated modules, and effective final product.
8. Verification is done by QA team to ensure that the software is as per the specifications in the SRS document.	8. Validation is carried out with the involvement of testing team.
9. It generally comes first-done before validation.	9. It generally follows after **verification**.

Testing Methodologies

Black box testing

White box testing

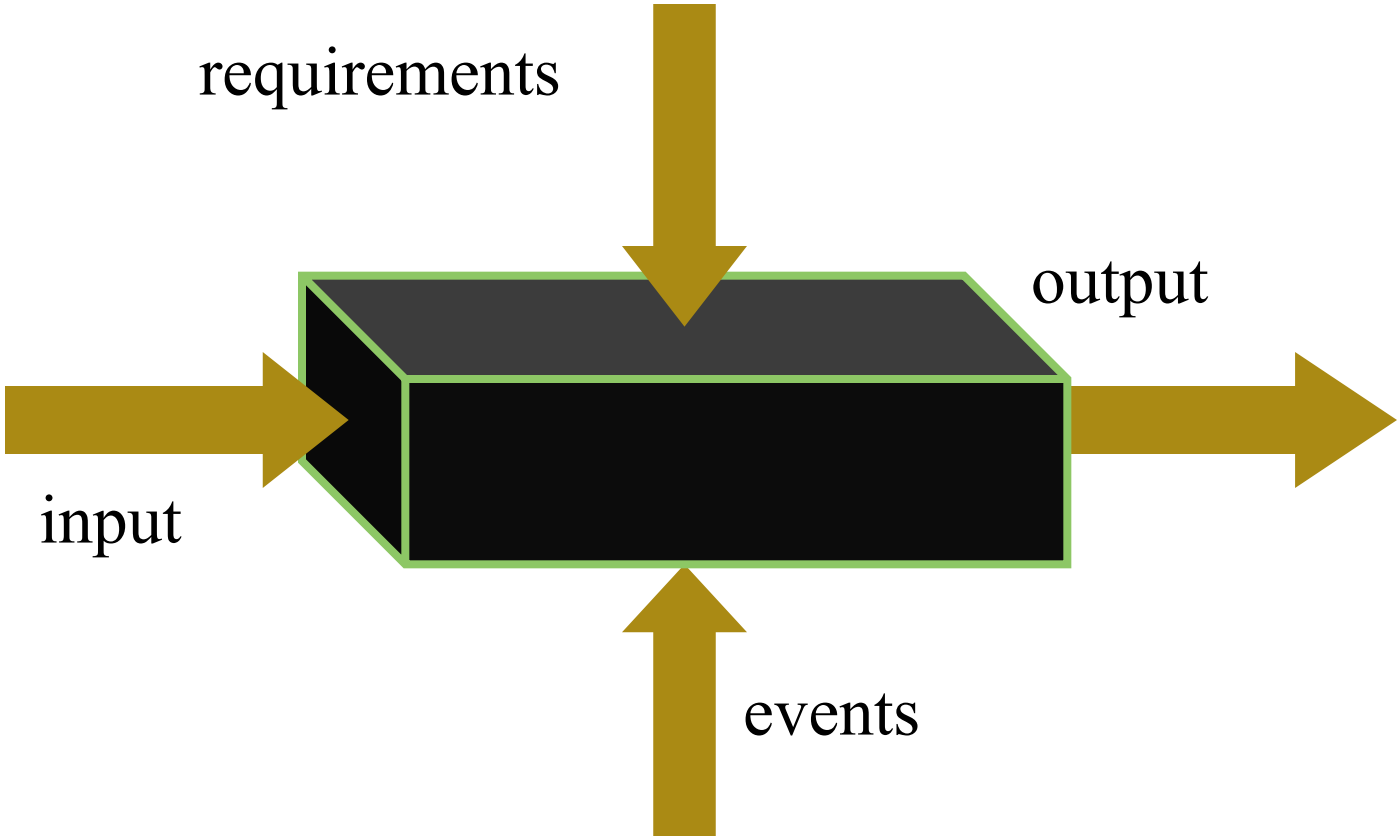
Black box testing

- No knowledge of internal program design or code required.
- Tests are based on requirements and functionality.

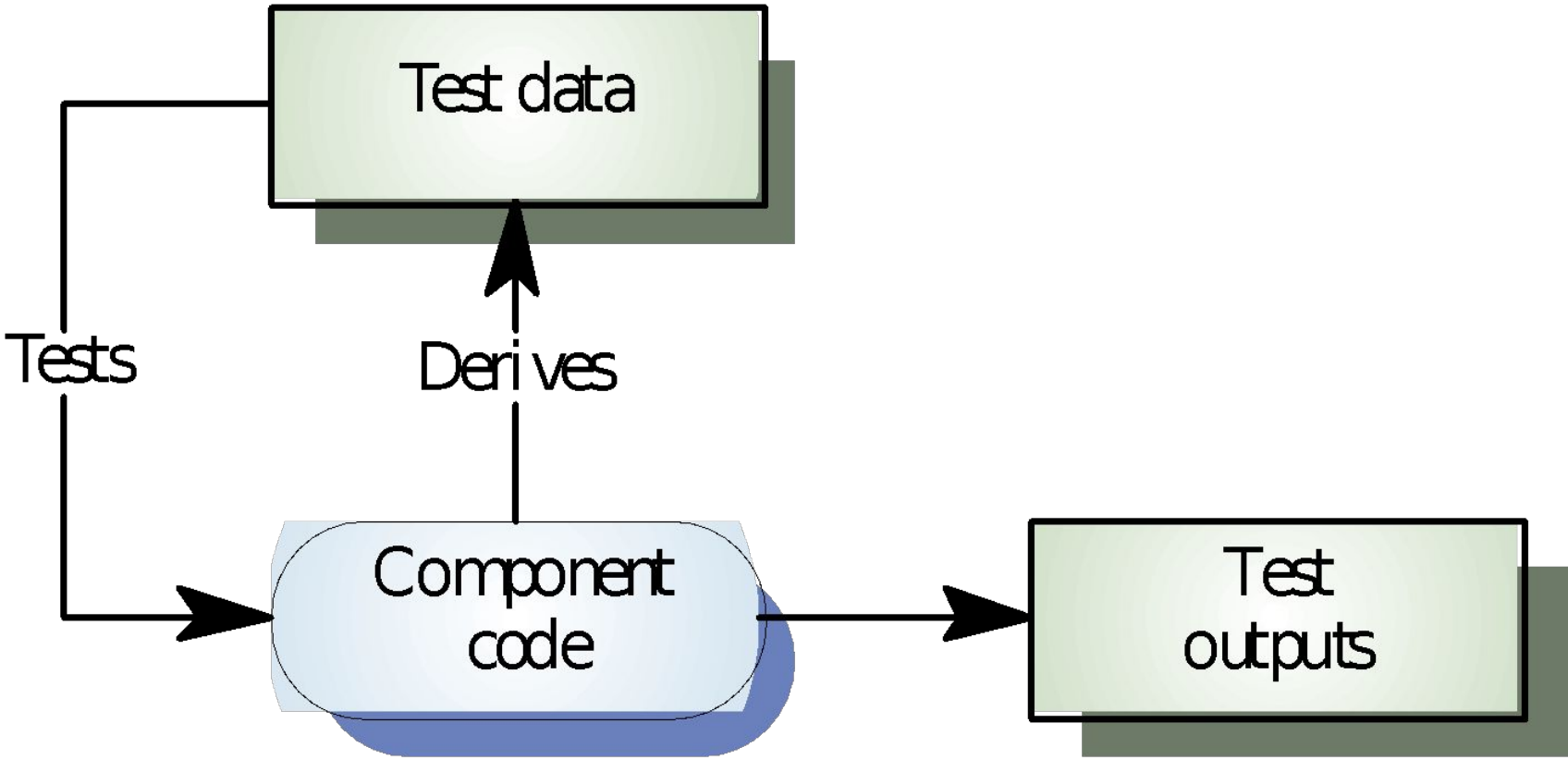
White box testing

- Knowledge of the internal program design and code required.
- Tests are based on coverage of code statements, branches, paths, conditions.

Black box testing



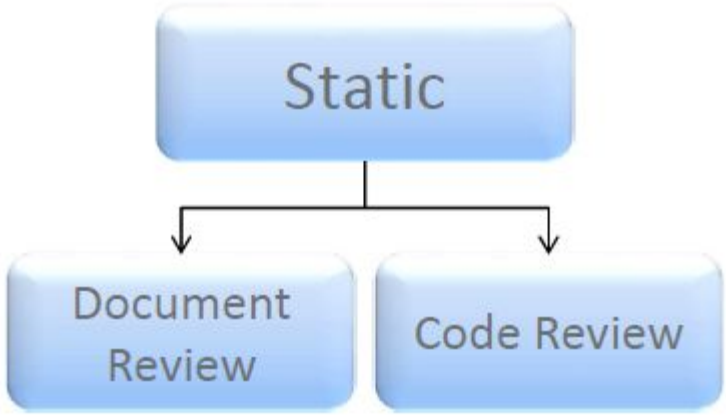
White box testing



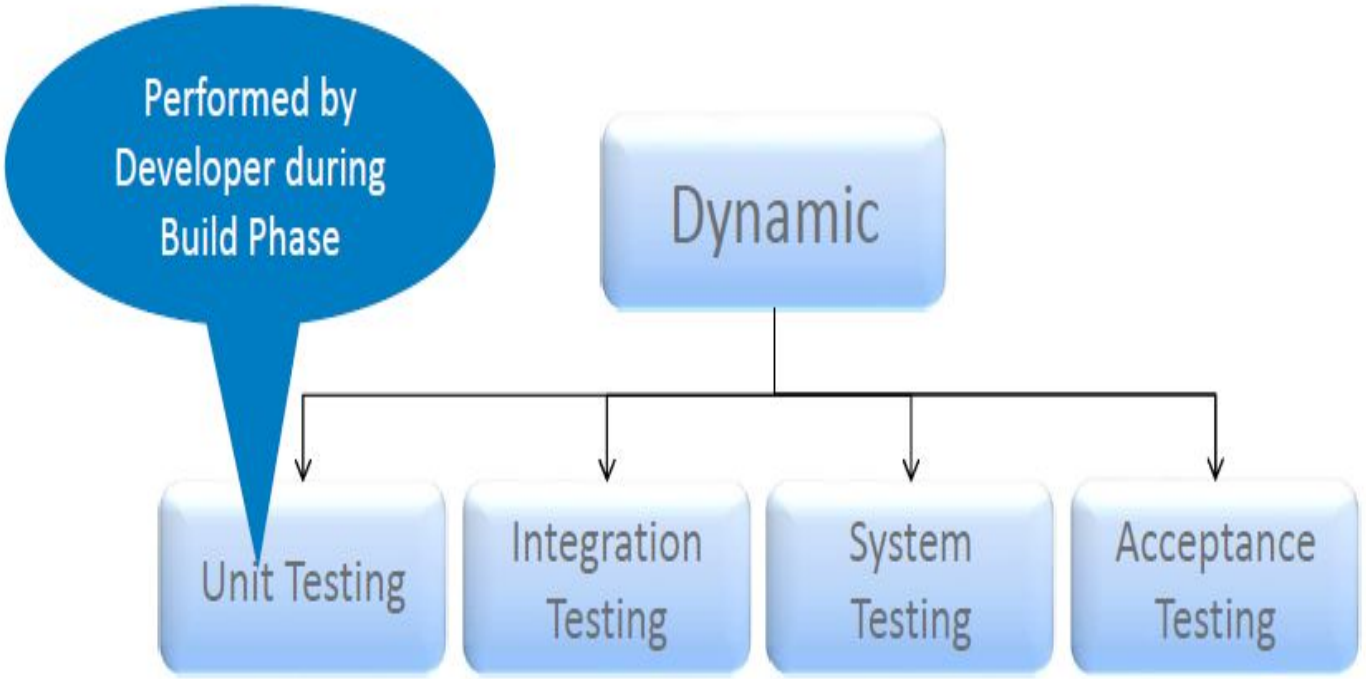
Types of Testing

- Static
- Dynamic

Types of Testing



Contd..


Performed by Developer during Build Phase
Dynamic
Unit Testing
Integration Testing
System Testing
Acceptance Testing

Static Testing

- Document Review
- Code Review

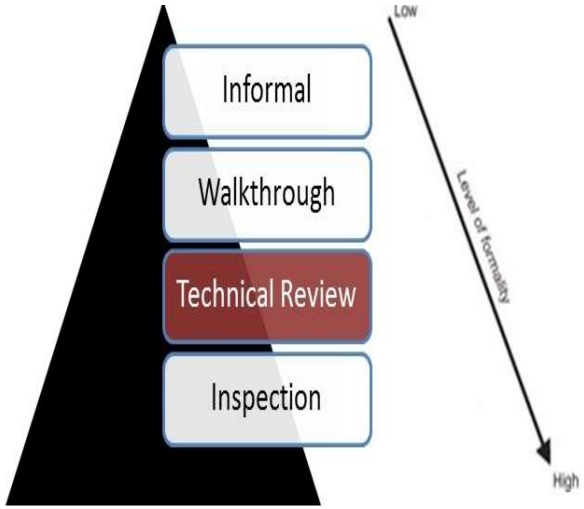
Static Testing Contd..

- Document Review
  - Document review is systematic examination (often as peer review) of any technical document or partial document, produced as a deliverable of a software development activity.
  - May include documents such as
    - contracts, project plans ,budgets
    - requirements documents, specifications, designs, source code
    - user documentation, support and maintenance documentation
    - test plans, test specifications, standards

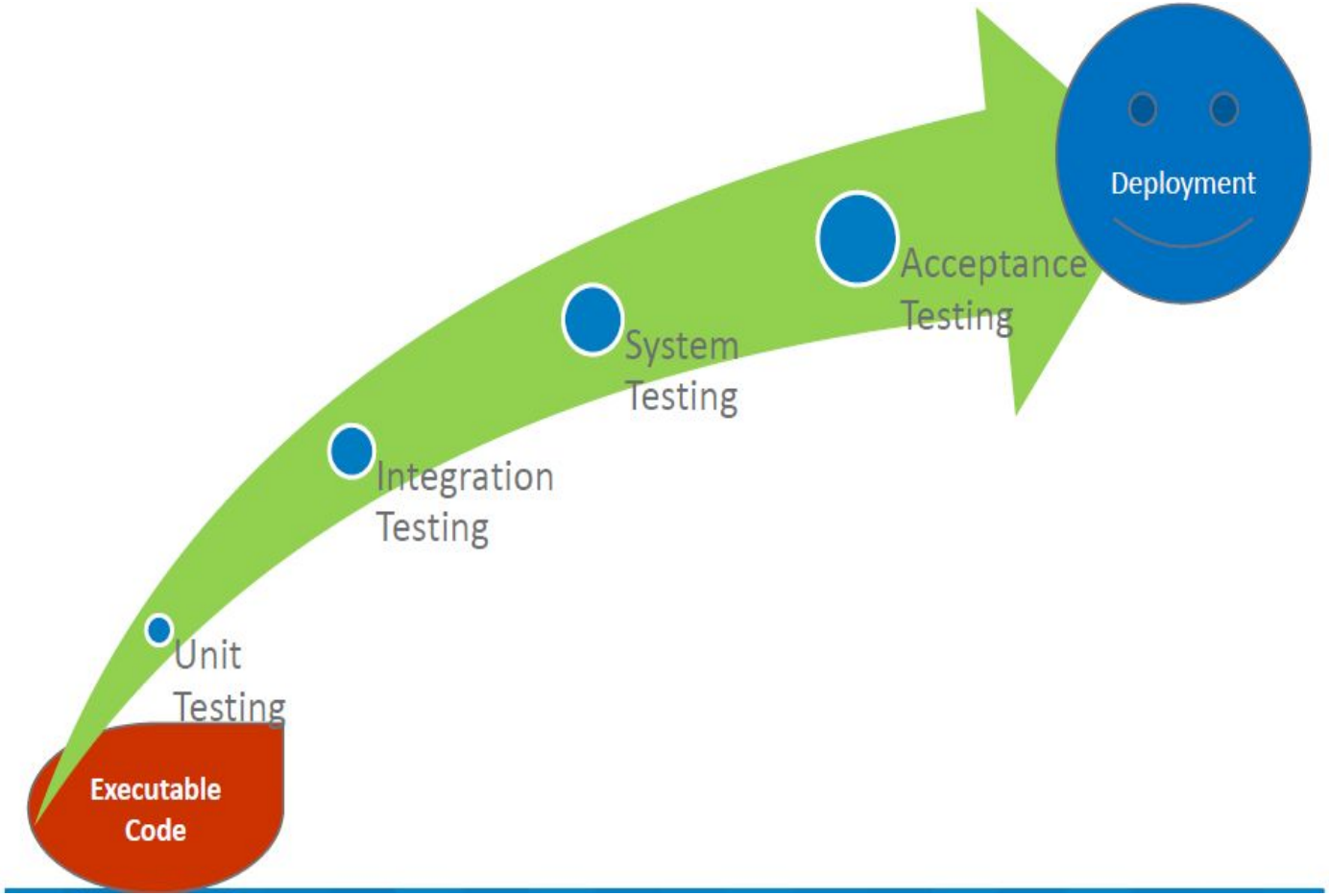
Static Testing Contd..

- Code Review

- It is a phase in the software development process in which the authors of code, peer reviewers, and perhaps quality assurance (QA) testers get together to review code.
- It is a systematic examination, which can find and remove the vulnerabilities in the code such as memory leaks and buffer overflows.
- It is systematic examination (often as peer review) of computer source code



Dynamic Testing
Levels of Testing:



Testing Levels

- Unit testing
- Integration testing
- System testing
- Acceptance testing

UNIT TESTING

- A unit is the smallest testable part of an application like functions, classes, procedures, interfaces.
- Unit testing is a method by which individual units of source code are tested to determine if they are fit for use.
- Tests each module individually.
- Follows a white box testing (Logic of the program).
- Done by developers.

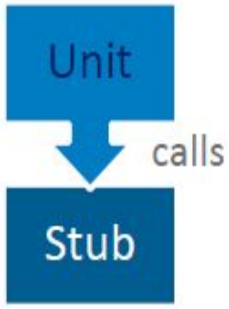
INTEGRATION TESTING

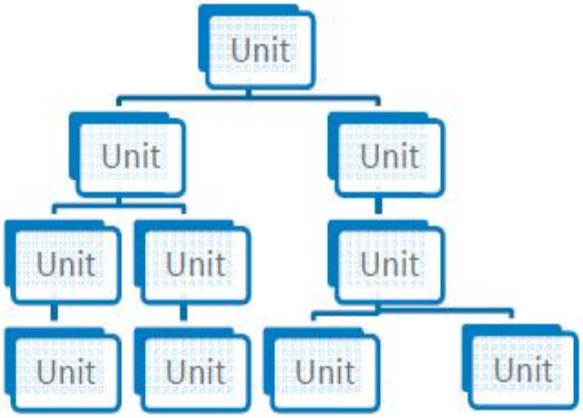
- Once all the modules have been unit tested, integration testing is performed.
- It is systematic testing.
- Produce tests to identify errors associated with interfacing.

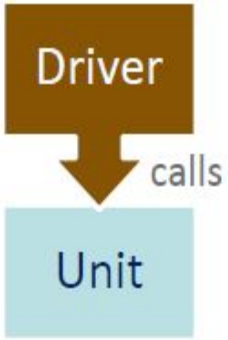
Integration Testing

Integration Testing

- In top down testing, top level modules are developed and tested first by using a dummy bottom level module called “Stub”
- In bottom up testing, bottom level modules are developed and tested first by using a dummy top level module called “Driver”



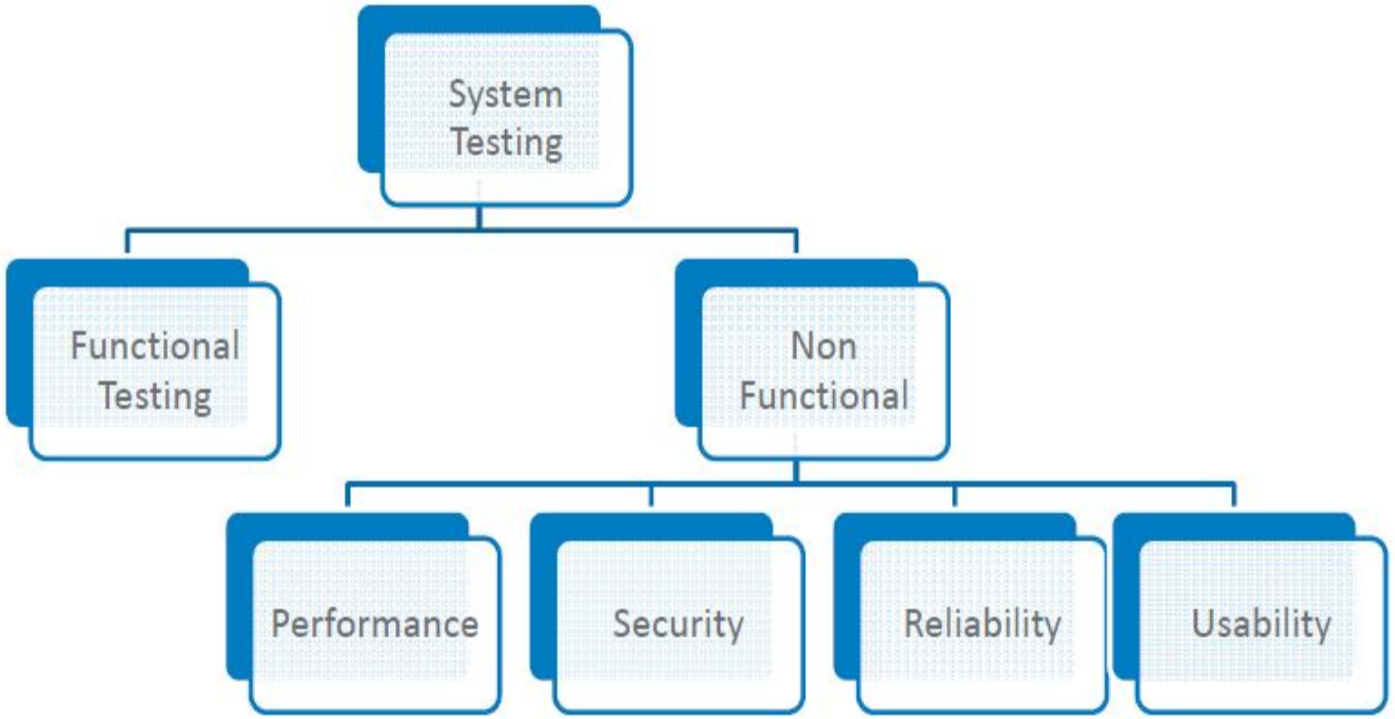




SYSTEM TESTING

- software testing where a complete and integrated software is tested.
- The purpose of this test is to evaluate the system's compliance with the specified requirements.

SYSTEM TESTING



SYSTEM TESTING

Types:

- Alpha Testing
- Beta Testing
- Acceptance Testing
- Performance Testing

- **Alpha Testing**

  It is carried out by the test team within the developing organization .

- **Beta Testing**

  It is performed by a selected group of friendly customers.

- **Acceptance Testing**

  It is performed by the customer to determine whether to accept or reject the delivery of the system.

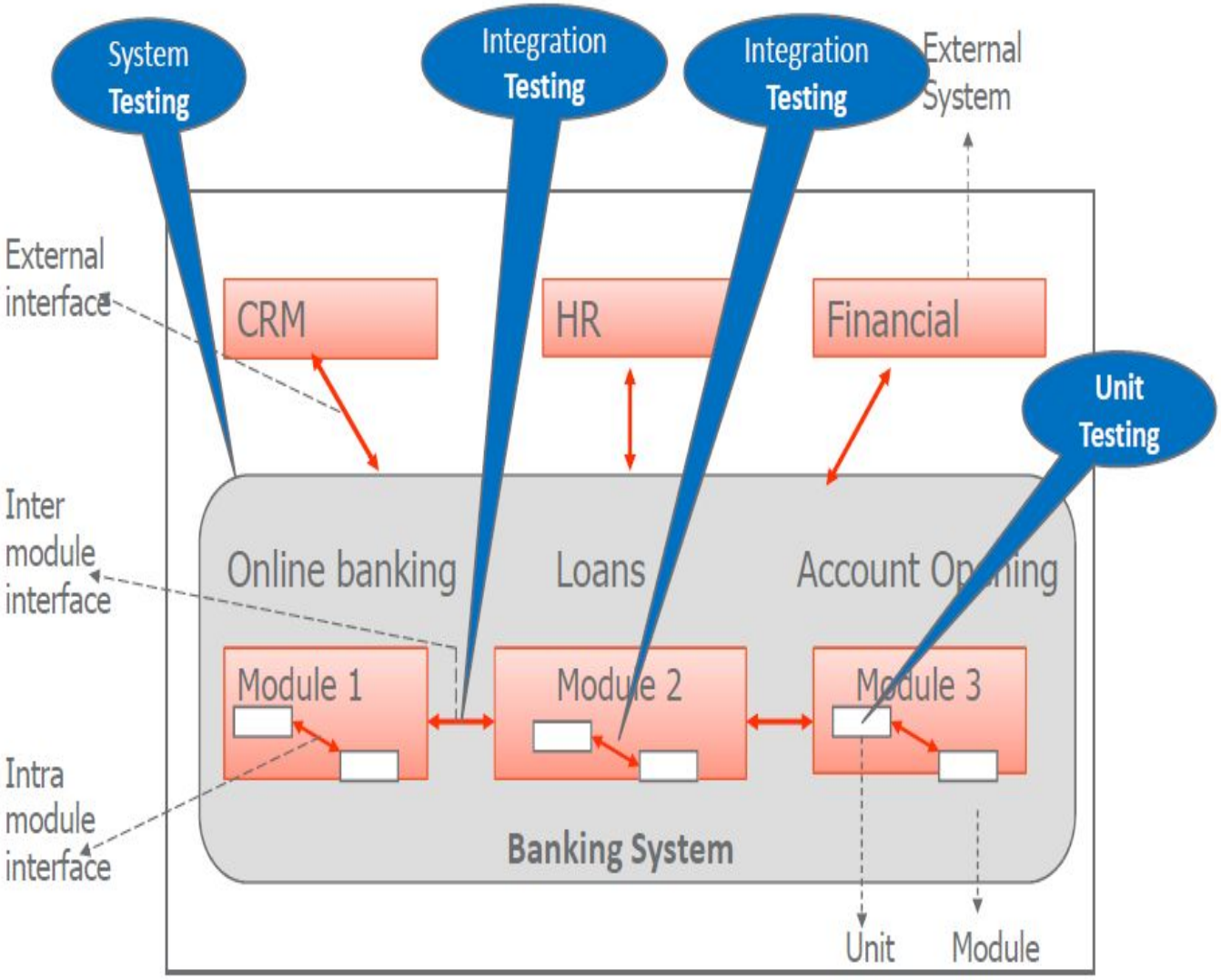
- **Performance Testing**

  It is carried out to check whether the system meets the nonfunctional requirements identified in the SRS document.

Types of Performance Testing:

- Stress Testing
- Volume Testing
- Configuration Testing
- Compatibility Testing
- Regression Testing
- Recovery Testing
- Maintenance Testing
- Documentation Testing
- Usability Testing

Levels of Testing (Contd..)



Levels of Testing (Contd..)

Level	What for?	Who does?	Where?
Unit Testing	Tests whether a particular module is meeting the detailed design specification	Developer	Developer's environment
Integration Testing	Tests whether the integrated modules are meeting the high level design specification	Development Team and Test Team	Testing environment
System Testing	Tests the system for functional and non-functional requirements	Test Team	Testing environment
Acceptance Testing	Tests whether the system tested software meets all the user requirements	Customer	Customer's environment

The following tools can be used for automation testing:

Software Testing Tools:

- The following tools can be used for automation testing:
- HP Quick Test Professional
- Selenium
- IBM Rational Functional Tester
- SilkTest
- TestComplete
- Testing Anywhere
- WinRunner
- LaodRunner
- Visual Studio Test Professional

Thank You !

Questions?