VIT-AP UNIVERSITY

Module – 1

Software Testing (SWE4002)

Module No. 1	Software Testing	8 Hours
Software Testing: Introduction, Evolution, Myths & Facts, Goals, Psychology, Definition, Model for testing, Effective Vs Exhaustive Software Testing. Software Testing Terminology and Methodology: Software Testing Terminology, Software Testing Life Cycle, relating .		

Prepared By: Dr. Radha Mohan Pattanayak

What is Testing?

Testing is the process of determining if a program behaves as expected. In this process one may discover errors in the program under test.

What is Debugging?

After reveals the error by the testing process of determining the cause of the error and find solution for it is called debugging.

What is Software Testing?

It is a static or dynamic process to test the different stages of the software to find the bugs in it before delivery of it to the customer.

Roles of Software Testing

- ✓ Testing is a integral part of development to find out the pitfalls
- ✓ When Testing find defects, those defect are required to improve the quality of software
- ✓ It ensures to deliver the quality product to the end user.

Why Software Testing?

- ✓ To improve the quality of the product
- ✓ To minimize the risk of failure
- ✓ To deliver a quality product
- ✓ To make sure that the end result meets the requirements

Software Myth and facts

What is Myth?

- ✓ Myth represents a wrong belief or wrong information about a particular object or situation.

What is S/W Myth ?

- ✓ S/W myth represents a wrong belief or wrong information about the S/W and the process behind to built the S/W.

S/W Myths are categorized into mainly three different types as:

1) Management myth

2) Customer myth

3) Developer myth

1) **Customer myth**: **In general the S/W myth believed by customer who can be either internal or external.**

   - ✓ Customers believe that general statement of objectives is sufficient to begin writing program
   - ✓ We can add more program to catch up
   - ✓ They think they have latest computers

2) **Management myth**

   - ✓ Standard tools are present, they are sufficient for developers
   - ✓ They think they have latest computers and they are sufficient for developers
   - ✓ A good manager can manage any type of projects

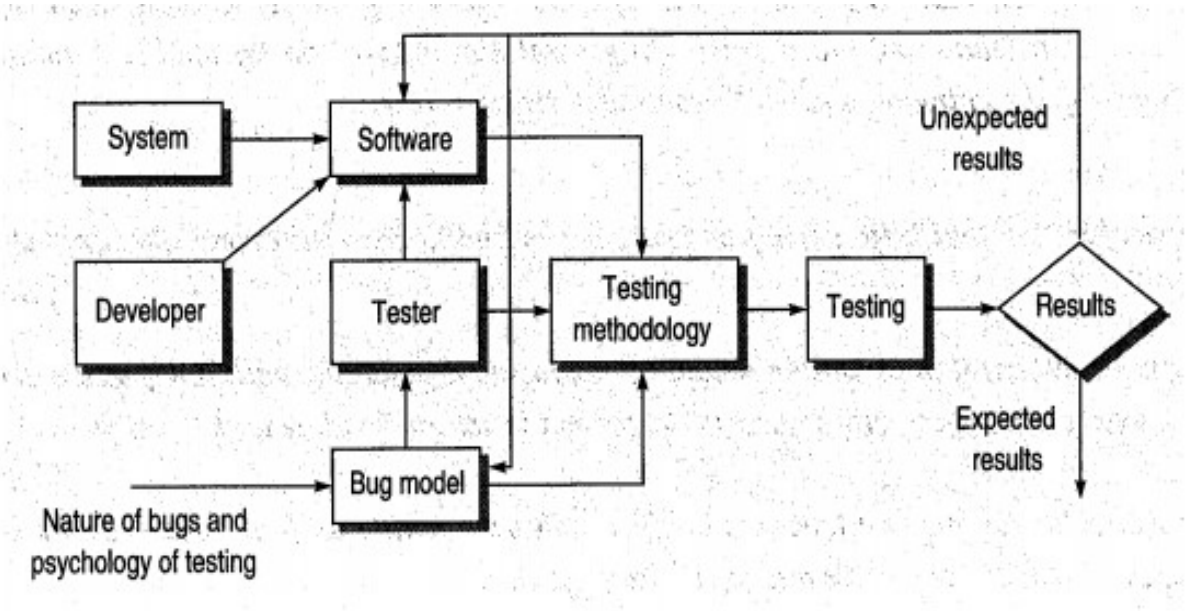
3) **Developer myth**

   - ✓ If I miss something now, I can fix it later
   - ✓ Once the program is written and running, then my work completed
   - ✓ Until the program is running there is no way of accessing it's quality
   - ✓ The only deliverable thing for a S/W project is working program

Software Model:

The model for testing process mainly consists of three models as.

- ❑ A model of the environment
- ❑ A model of the Tester and Testing methodology
- ❑ A model of the expected bugs



A model of the environment:

- ✓ A program's environment is the hardware and software required to make the software run.
- ✓ The environments also includes all the programs like OS, loader, linker, compiler etc. which are used for the smooth execution of the software.
- ✓ For online systems, the environment may include communication lines, other systems, terminals and operators.
- ✓ The software is built after analysing the system in the environment.
- ✓ The concept of the program is to be simplified in order to test it.
- ✓ The developer should design and code the software such that it is testable at every point, thus avoiding unnecessary complexities.

A model of the Tester and Testing methodology

- ✓ Based on the inputs from the software model and the bug model, testers can develop a testing methodology that incorporates both the testing strategy and testing tactics.
- ✓ If simple model of the program are unable explain the unexpected behaviour, then the model need to modify with more facts and in detail information.
- ✓ And after it also it fails then it needs to modify the program.

A model of the expected bugs:

- ✓ Bug model provides a perception of the kind of bugs expected.
- ✓ Considering the nature of all types of bugs, a bug model that may help in deciding a testing strategy can be prepared.
- ✓ However, every type of bug cannot be predicted. Therefore, if we get incorrect results, the bug model needs to be modified.

Exhaustive software testing

- ❑ Exhaustive software testing is the process of testing the functionality of the software where all possible inputs and their combinations are run along with different preconditions.
- ❑ Exhaustive software testing ensures there are no undiscovered faults at the end of the test phase.

Effective software testing

- ❑ Effective software testing provides experienced-based practices and key concepts that can be used by an organization to implement a successful and efficient testing program.
- ❑ It tests the effectiveness of an application and tests particular functions.

Difference between Exhaustive & Effective S/W testing

Exhaustive Testing	Effective Testing
✓ Test the software for all possible input values.	✓ It tests the effectiveness of an application and tests particular functions for given resources.
✓ It is practically not possible.	✓ It is possible to test the system.
✓ It is very exhaustive and time-consuming.	✓ It does not take much time for the testing team.
✓ Mostly it is documented and theoretical approach.	✓ It is a completely practical approach and the tester test the effectiveness of the system.
✓ It is costly process due to exhaustive nature of checking all test cases.	✓ It is economic and does not extend the budget boundaries of the customer.
✓ It is complete testing and covers all possible test scenarios.	✓ It prioritizes test scenarios and smart testing techniques.

Software testing Terminologies

What is a bug?

- ✓ In software testing, a bug is the informal name of defects, which means that software or application is not working as per the requirement.
- ✓ When we have some coding error, it leads a program to its breakdown, which is known as a bug. The test engineers use the terminology Bug.

What is a Defect?

- ✓ When the application is not working as per the requirement is knows as defects.
- ✓ In other words, we can say that the bug announced by the programmer and inside the code is called a Defect.

What is a Error?

- ✓ The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly.

What is a Fault?

- ✓ The fault may occur in software program because of the following reasons:
  - ➢ Lack of resources
  - ➢ An invalid step
  - ➢ Inappropriate data definition

What is a Failure?

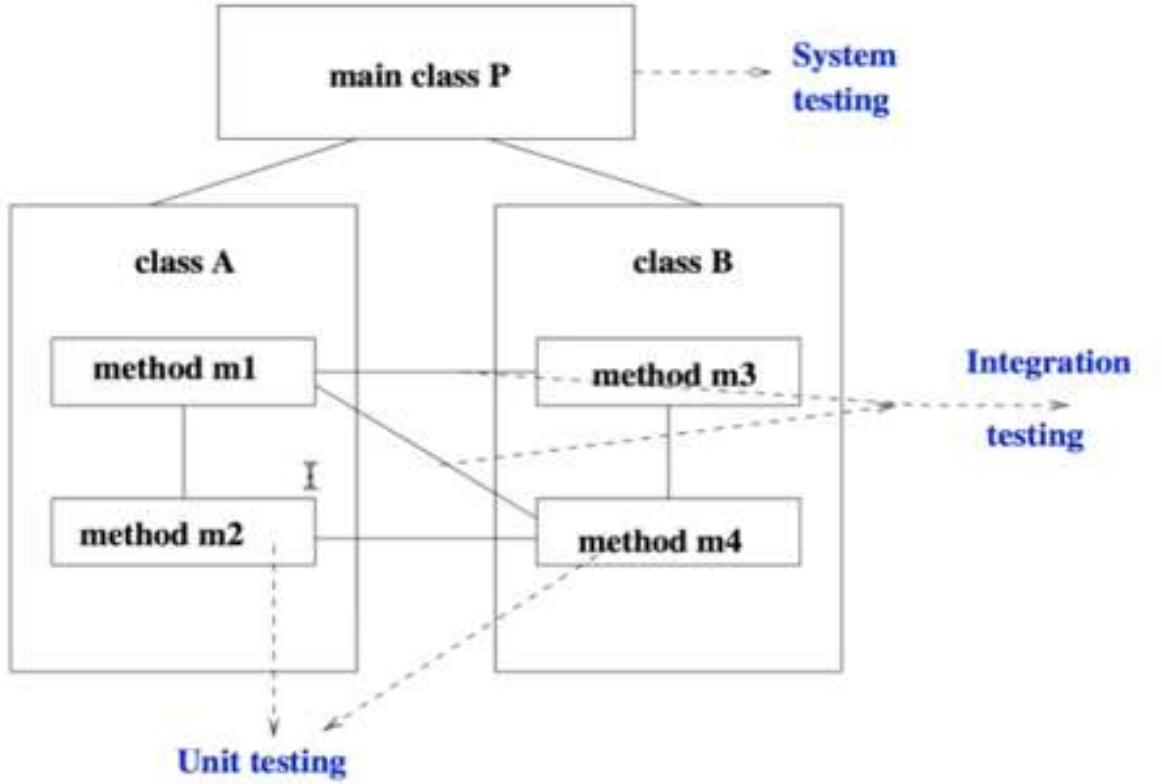
- ✓ The defects lead to a loss specifies a fatal issue in software/ application or in its module, which makes the system unresponsive or broken is called as failure.

In other words, we can say that if an end-user detects an issue in the product, then that particular issue is called a failure.

- ✓ A test case is typically involves inputs to the software and expected outputs.
- ✓ When the test cases are executed and results recorded, if the actual output matches the expected output, then the test case is passed otherwise it is said the test case is failed.
- ✓ Normally a failed test case indicates error.
- ✓ Sometimes a test case also contains other parameters like test case ID, traceability details etc. (Traceability means who, where, what functionality are going to test)

In the various phases of software development we are performing various levels of tests to major the productivity and efficiency of the software model. The different test may be….

- **Unit testing:** This test has done by the developer for each module during the coding process.
- **Integration testing:** In this method various components are put together and perform the testing.
- **System testing:** Done with full system implementation and platform in which the system will be running.



- **Acceptance testing:** It has done by the end user to ensure that the delivered products meet the committed requirements or not.
- **Beta testing:** This type of testing is done by the end user on the beta version of the software after it's release.
- **Functional testing:** This type of testing has been done to ensure weather the software meets its specified functionality.
- **Stress Testing:** This type of testing is done t evaluate how the system behaves under the peak or unfavourable conditions or inputs.

- **Performance testing:** This type of testing is done to major the speed or response time of the system.
- **Regression testing:** Mainly this type of testing is done after modifying or upgrading to any component in the software model, to ensure that the modification has performed accurately with out disturbing to other components in the model.

White Box Testing	Black Box Testing
✓ To perform testing we should understand the programming language	✓ To perform testing there is no need to have an understanding of the programming languages
✓ In this we will look into the source code and test the logic of the code	✓ In this we will verify the functionality of the application based on requirement
✓ In this the developer should know the internal structure of the code	✓ In this no need to know the internal design of the code
✓ Can be implemented mainly at unit testing as well as in integration and system testing.	✓ Can be implemented at unit testing, in integration testing, system testing as well as acceptance testing.
✓ The developers can perform the white box testing	✓ The test engineer will perform black box testing

Software Testing Life Cycle:

- STLC identifies what test activities to carry out and when to accomplish those test activities.
- Every testing process has gone through different stages is called Software Testing Life Cycle.

Different phases in STLC

- ✓ Requirement Analysis
- ✓ Test planning
- ✓ Test design
- ✓ Test environment set up
- ✓ Test execution
- ✓ Test case closure.

Requirement Analysis:

- ✓ During this phase, the test team studies and analyse the requirements from a testing perspective.
- ✓ This phase helps to identify whether the requirements are testable or not.
- ✓ During this phase if any requirement is not testable, the test team can communicate with various stakeholders such as client, business analysis, technical leads, system architects etc. so that the changed strategy can be planned accordingly.

Test planning:

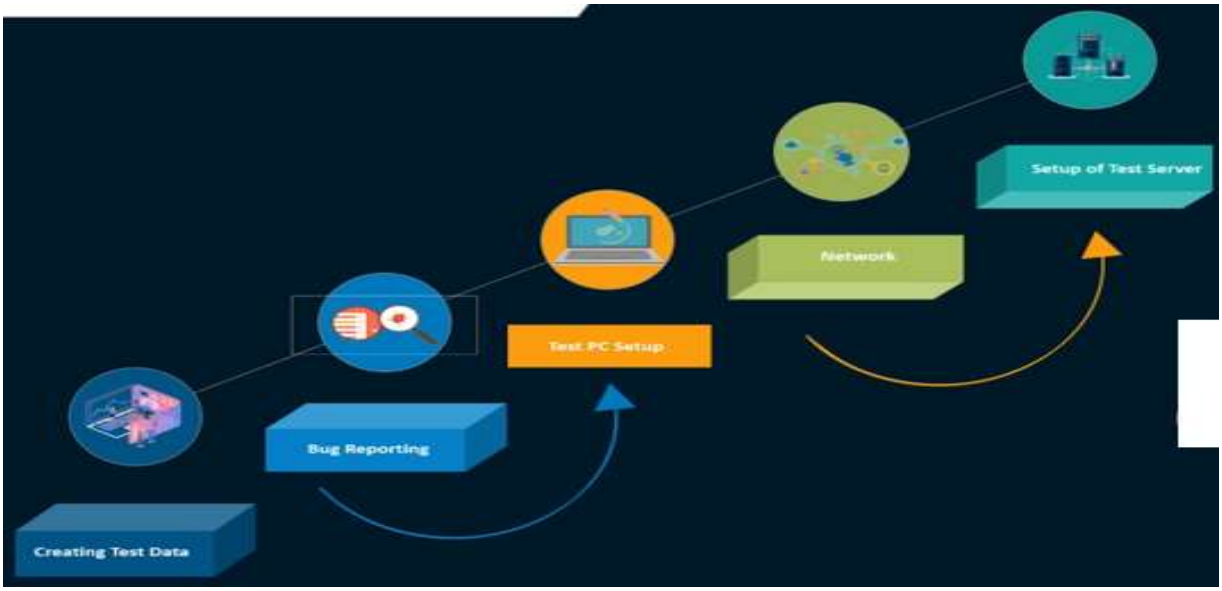
- ✓ It is the first step of the testing process.
- ✓ Analyse the product: This step represents you must learn the product thoroughly before it's release.
- ✓ The test manager or test lead involves to determine the effort and cost estimates for entire project.
- ✓ Preparation of test plan will be done based on the requirement analysis.
- ✓ Activities like resource planning, training requirements, determining roles and responsibilities carried out in this phase.

Design Test strategy:

- ✓ The test case development activity is started once the test planning activity is finished.
- ✓ Test strategy document is a high level important document which is developed by test manager. Generally this document represents the project testing objectives and the means to achieve them.
- ✓ This is the phase of STLC where testing teams start with test cases design. Test team prepares test cases, test scripts and test data.
- ✓ Once the test cases are ready then these test cases are reviewed by team lead.
- ✓ Besides from this the test team prepares the requirement traceability matrix known as RTM which verifies weather the requirements of the end-user are fulfilled or not.

Intention to develop a good test cases

- ✓ Test cases need to be very simple
- ✓ Create test case with based on end user requirements
- ✓ Avoid assumptions
- ✓ Test cases must be identifiable
- ✓ Avoid repeatation of test case and ensure 100% coverage.



- ✓ In this phase the test team will analyse the requirements and prepare the list of software and hardware required.
- ✓ Some times the testers have no work in the setting development, at that time they will go for smoke test cases.
- ✓ Smoke testing is a software testing technique intended to ensure that essential features of a software application are functioning correctly. A smoke test is designed to identify any major issues that would prevent further testing of the software.

Test Execution:

- ✓ Testing team starts executing test cases based on prepared test planning as per the priority based.
- ✓ After execution mark the status of test cases.
- ✓ If any test case is failed then corresponding defect is assigned with a Bug ID and reported to developer team via bug tracking system.
- ✓ If any of the test cases are blocked due to any defect then such test cases can be marked as blocked and reported to developer team.

Test Closure Report:

- ✓ Test Summary report
- ✓ Identifier
- ✓ Test summary
- ✓ Comprehensive Assessment

- ✓ Summary of Results
- ✓ Evaluation
- ✓ Summary of activities
- ✓ Approval

Software Development Life Cycle

Feasibility Study:

- Feasibility study involves analysis of the problem and it is the most important and fundamental stage.
- In a software company, even customer comes with the problem and we analyse that we are feasible or not to solve the customer's problems.
- We estimate the rough idea of the resource requirement as well as estimate time for completion.

There are five different types of feasibilities checks:

- ✓ Economic: Can we complete the project within the budget or not?
- ✓ Legal: Can we handle this project as cyber law and other regulatory frame work.
- ✓ Operation feasibility: Can we create operations which is expected by the client?
- ✓ Technical: Need to check whether the current computer system can support the software.
- ✓ Schedule: Decide that the project can be completed within the given schedule or not.

Requirements gathering and Analysis:

- ✓ The goal of requirement gathering activity is to collect all relevant information from the customer regarding the product to be developed. This is done to clearly understand the customer requirements so that incompleteness and inconsistency are removed.
- ✓ The requirement analysis activity is begun by collecting all relevant data regarding the product to be developed from the users and customer through interviews and discussion.

Requirements Specification or documentation:

- ✓ After all ambiguities, inconsistencies, and incompleteness have been resolved and all the requirements properly understood, the requirements specification activity can start.

- ✓ Requirement Documentation is very important activity after the requirement gathering and analysis. This is a way to present requirements in a consistent format.
- ✓ During this activity, the user requirements are systematically organized into a software requirements specification(SRS) documents.

A specification document can be:

- ✓ A written document
- ✓ A graphical model
- ✓ A formal mathematical model
- ✓ A collection of usage scenarios
- ✓ A prototype

Design document:

- ✓ The goal of design phase is to transform the requirements specified in the SRS documents into a structure that is suitable for implementation in some programming languages.
- ✓ In this phase, we design the architecture of the system using the SRS documents.
- ✓ The system design is the process of designing the architecture, components, modules, interface for a system to satisfy specified requirements.

There are two types of design documents developed in this phase as High level design and Low level design.

High level design:

- ✓ Brief description and name of each module.
- ✓ An outline about the functionality of every module
- ✓ Interface relationship and dependencies between modules
- ✓ Database tables identified along with their key elements
- ✓ Complete architecture diagrams along with technology details.

Low level design:

- ✓ Functional logic of the modules
- ✓ Database tables which includes type and size
- ✓ Complete details of the interface
- ✓ Address all types of dependency issues

- ✓ Listing of error message
- ✓ Complete input and outputs for every module.

Coding document:

- ✓ This is the longest phase of the SDLC process.
- ✓ In this phase the developers start built the entire system by writing the codes considering a small unit or module wise.
- ✓ In this phase, the tasks are divided into unit or module wise and assigned to each developers.

Software Maintenance:

- ✓ Maintenance is the backbone of the software success.
- ✓ Software maintance represents, the modification of a software product after delivery t correct faults, to improve performance or other attributes, r to adopt product to a modified environment.