5... Software Testing

Learning Objectives...

- To learn the concept of Software testing.
- To study the Types of software testing.
- To understand the concept of Verification and Validation.
- To learn about Black box and White box testing.

5.1 INTRODUCTION

According to **ANSI/IEEE 1059** standard – A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

1.1 Need/Necessity of Testing

Software testing is essential to ensure that software products meet the expected requirements, function correctly under all conditions, and are free of defects. The main purposes of software testing include:

- **Quality Assurance:** Verifies that the product meets quality standards.
- **Error Detection:** Identifies bugs or issues before deployment.
- **Cost Reduction:** Fixing defects early in the development process is cheaper tha after release.
- **Security:** Ensures that software does not have vulnerabilities.
- **User Satisfaction:** Delivers a reliable and user-friendly product.

2 Testing Terminology

Iere are key terms used in software testing:

Test Case: A set of actions and inputs to verify a specific function of the software

Bug/Defect: An error found in the software.

- Test Plan: A document outlining the scope, approach, resources, and schedule of testing.
- **Test Suite:** A collection of test cases that are intended to be used together.
- **Test Environment:** The setup of software and hardware for the testing process.
- **Regression Testing:** Re-running tests after changes to ensure existing functionality works.
- **Verification:** Ensuring the product is built correctly (meets design specs).
- **Validation:** Ensuring the right product is built (meets user needs).

5.2 DEFINITION OF SOFTWARE TESTING

- **Software Testing** is the process of evaluating and verifying that a software application or system performs its intended functions. It involves executing the software with the intent to find errors and ensure the product is of high quality.

5.2.1 Life Cycle of Software Testing

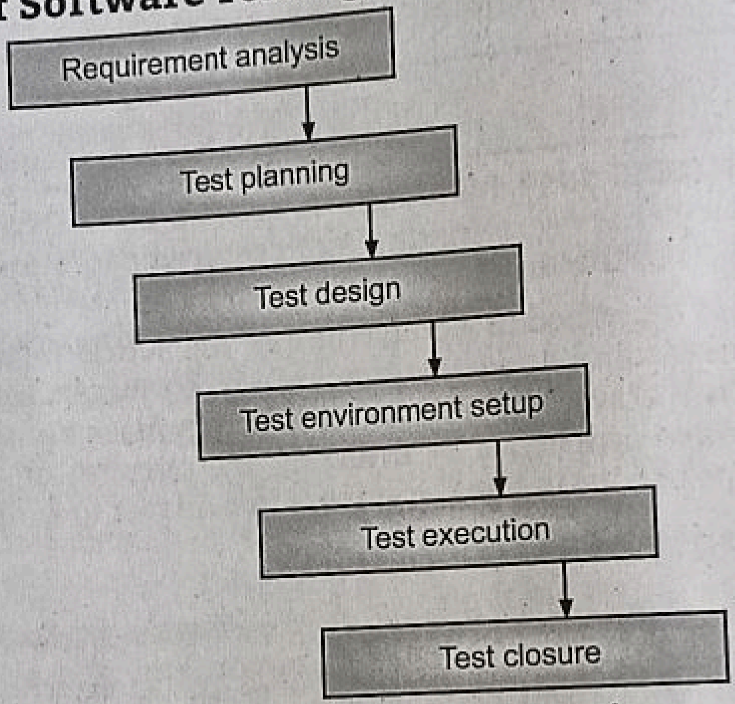

Fig. 5.1: Life Cycle of Software Testing

1. **Requirement Analysis:**
   - Understand what needs to be tested.
   - Identify testable requirements.
2. **Test Planning:**
   - Define scope, strategy, resources, and schedule.
   - Create a test plan document.
3. **Test Case Development:**
   - Design and write test cases.
   - Prepare test data.
4. **Test Environment Setup:**
   - Configure the hardware and software for testing.

5. **Test Execution:**
   - Run test cases and log defects.
6. **Test Closure:**
   - Evaluate exit criteria.
   - Prepare test summary reports and learn from the outcomes.

5.2.2 Types of Testing

(a) Manual Testing:

- Manual Testing is the process of manually executing test cases without the use of automation tools. Testers play the role of end users and use the software to find unexpected behavior or bugs.

Characteristics:

- No scripts or automation tools used.
- Tester prepares test cases, executes them, and reports defects.
- Used for exploratory, usability, and ad-hoc testing.

Advantages:

1. Useful for small projects or early development stages.
2. Better for detecting visual and usability issues.

Disadvantages:

1. Time-consuming and labor-intensive.
2. Less reliable for repeated testing compared to automation.

(b) Automation Testing:

- Automation Testing involves using software tools and scripts to execute tests automatically, compare actual outcomes with expected outcomes, and report the results.

Characteristics:

- Tests are executed by tools (e.g., Selenium, QTP, TestNG, JUnit).
- Once written, tests can be reused and run quickly across versions.

Advantages:

1. Faster execution of tests (especially regression tests).
2. Reduces human error.
3. Supports continuous integration/continuous deployment (CI/CD) processes.

Disadvantages:

1. Initial setup is time-consuming and costly.
2. Not suitable for all types of testing (e.g., exploratory testing).
3. Requires programming knowledge.

Examples of Automation Tools:

- Selenium (Web applications)
- Appium (Mobile applications)
- JUnit/TestNG (Java-based unit testing)
- Cypress (Modern front-end apps)

5.3 VERIFICATION AND VALIDATION

- These are two important concepts in quality assurance:
  1. **Verification:**
     - *"Are we building the product right?"*
     - A process of evaluating intermediate work products of a software development lifecycle to ensure that we are building the product correctly.
     - **Activities:** Reviews, walkthroughs, inspections.
  2. **Validation:**
     - *"Are we building the right product?"*
     - The process of evaluating the final product to ensure it meets business needs and requirements.
     - **Activities:** Testing the actual product (system testing, UAT).

5.4 BLACK BOX TESTING

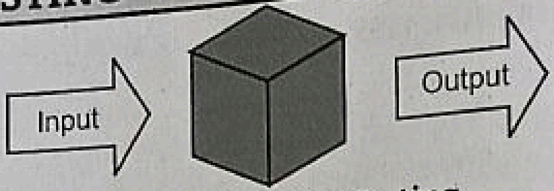

Fig. 5.2: Black Box Testing

- **Black Box Testing** is a testing approach where the tester does not need to know the internal code or structure of the application.

Focus:

- Inputs and expected outputs.
- Functionality of the system as a whole.

Techniques Used:

- **Equivalence class:** The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- **Boundary values:** The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- **Cause-effect graphing:** In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- **Pair-wise Testing:** The behavior of software depends on multiple parameters. In pair-wise testing, the multiple parameters are tested pair-wise for their different values.
- **State-based testing:** The system changes state on provision of input. These systems are tested based on their states and input.

Advantages:

1. Tester does not need programming knowledge.
2. Tests are from the user's perspective.

Disadvantages:

1. Limited coverage since internal logic is not tested.
2. Hard to identify some types of bugs.

5.5 WHITE BOX TESTING

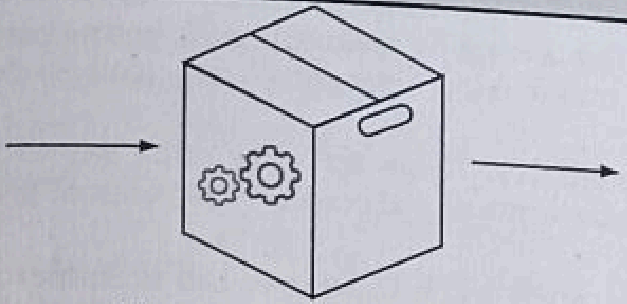

Fig. 5.3: White Box Testing

- **White Box Testing** (also known as structural testing or glass-box testing) involves testing the internal structures or workings of an application.

Focus:

- Logic, paths, and internal code structure.

Techniques Used:

1. Statement Coverage
2. Branch Coverage
3. Path Coverage
4. Loop Testing

Advantages:

1. Thorough testing of internal logic.
2. Detects hidden software bugs effectively.

Disadvantages:

1. Requires knowledge of programming.
2. Time-consuming for large systems.

5.6 TESTING LEVELS

- Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.
- Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels that discussed below:

1. **Unit Testing:**

- While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

2. **Integration Testing:**

- Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updation etc.

3. **System Testing:**

- The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:
  - **Functionality Testing:** Tests all functionalities of the software against the requirement.

- **Performance Testing:** This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
- **Security and Portability:** These tests are done when the software is meant to work on various platforms and accessed by number of persons.

4. **Acceptance Testing:**

- When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.
  - **Alpha Testing:** The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.
  - **Beta Testing:** After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

5. **Regression Testing:**

- Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

Check Your Understanding

1. Why is software testing necessary?
   (a) To write code faster
   (b) To ensure the software meets the requirements and is defect-free
   (c) To reduce the team size
   (d) To increase software complexity
2. In testing terminology, a defect is:
   (a) A hardware failure
   (b) A missing feature
   (c) A deviation from expected behavior
   (d) A function call
3. What is meant by a test case?
   (a) The source code of the application
   (b) A set of input values, execution conditions, and expected results
   (c) A software requirement
   (d) A user manual
4. Software Testing is defined as:
   (a) The process of identifying errors in a software system
   (b) The process of developing new features
   (c) Installing the software
   (d) Designing databases

5. Which of the following is the correct order in the Software Testing Life Cycle (STLC)?
   (a) Planning → Execution → Reporting → Analysis
   (b) Testing → Debugging → Deployment
   (c) Design → Coding → Maintenance → Testing
   (d) Requirement Analysis → Test Planning → Test Case Design → Execution → Closure

6. Manual testing is done by:
   (a) Automated tools (b) Hardware components
   (c) Testers without the use of tools (d) Robots

7. Which of the following is an advantage of manual testing?
   (a) Fast and repetitive (b) Easy to catch visual issues
   (c) Very accurate for large data sets (d) Cannot be changed

8. Automation testing is best suited for:
   (a) Repetitive test cases (b) One-time execution
   (d) Exploratory testing (d) Short-term projects

9. Which tool is commonly used in automation testing?
   (a) Paint (b) Notepad
   (c) MS Word (d) Selenium

10. Verification ensures:
    (a) That the product is being built right
    (b) That the right product is being built
    (c) That the product is sold
    (c) That the user is trained

11. Validation is concerned with:
    (a) Source code format (b) Meeting user requirements
    (c) Hardware performance (d) Compiler checks

12. Black Box Testing focuses on:
    (a) Code structure
    (b) Internal logic
    (c) Hardware bugs
    (d) Inputs and outputs without knowing internal code

13. Which of the following is a Black Box Testing technique?
    (a) Code walkthrough (b) Statement coverage
    (c) Equivalence partitioning (d) Path testing

14. White Box Testing focuses on:
    (a) User experience (b) Testing internal logic of the code
    (c) Hardware configuration (d) System documentation

15. Which of these is used in White Box Testing?
    (a) Syntax testing (b) Decision table
    (c) Path coverage (d) Load testing

Answers

1. (b)	2. (c)	3. (b)	4. (a)	5. (d)	6. (c)	7. (b)	8. (a)
9. (d)	10. (a)	11. (b)	12. (d)	13. (c)	14. (b)	15. (c)	

Practice Questions

Q.I Answer the following questions in short.

1. Define the following: defect, error, fault, bug, failure, test case, test suite.
2. What are the objectives of software testing?
3. What is:
   (a) Software testing
   (b) Manual testing
   (c) Automation testing
   (d) Black box testing
   (e) White box testing
   (f) System testing
   (g) Unit testing
   (h) Integration testing
   (i) Regression testing
4. Name popular tools used in automation testing (e.g., Selenium, QTP).
5. Define verification and validation.

Q.II Answer the following questions in detail:

1. Why is software testing important in the software development lifecycle?
2. Explain the need for testing in delivering a quality software product.
3. Differentiate between defect and failure.
4. Explain the Software Testing Life Cycle (STLC) with a diagram.
5. What are the various phases in STLC?
6. List advantages and disadvantages of manual testing.
7. Compare manual testing and automation testing.
8. Differentiate between verification and validation with examples.
9. At which stages are verification and validation applied?
10. What techniques are used in black box testing?
11. Give examples and advantages of black box testing.
12. When is black box testing most useful?
13. Explain white box testing techniques (e.g., statement coverage, branch coverage).
14. Compare black box and white box testing.
15. When is white box testing preferred?
16. Differentiate between verification and validation.
17. Write a short note on black box and white box testing.
18. What are the types of testing? Explain manual and automation testing.

***