

02-05-2024

# Software Testing Methodologies

## Assignment - 2

Bussa Sindhuja  
2103A51085  
Batch - B5

- 1 Outline the phases of the software testing life cycle (STLC) and describe the activities involved in each phase. How does adherence to the STLC contribute to the effectiveness of testing process?

The Software Testing life cycle (STLC) is systematic process used by software industry to design, develop and test high quality softwares. The SDLC aims to produce high quality softwares. The SDLC aims to produce high quality software that meets or exceeds customer expectations, reaches opinion completion within times.

The phases are:

Step 1 : Planning and Requirement Analysis :

It is most important performed by the senior members of the team with inputs from customer. Sales department, market surveys.

- Understand analyze the software requirements.
- Identify testing scope, objectives, constraints .
- Create a test plan outstanding test strategies, resources, schedules and deliverables.
- Establish metrics for measuring test effectiveness

Step 2 : Defining Requirements :

Once requirement Analysis is done the next step is to clearly define and document the product requirement and get them approved from the customer. This is done through software

requirement specification which consists of all product requirement to be designed.

### Step 3 : Designing the Product Architecture

SRS is reference for product architects to come out with the best architecture for the product to be developed. Based on requirements specified in SRS, more than one design approach for the product architecture is proposed. A design approach clearly defines all the architectural modules of product along with its communication.

### Step 4 : Building an Developing Product

In this stage SDLC the actual development starts and product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished.

### Step 5 : Testing the product

These stage is usually a subset of all stages as in modern SDLC. Models, the testing activities are mostly involved in stages of SDLC. This stage refers to the testing only stage of the product where product refers defects are reported, tracked, fixed and retested.

### Stage 6 : Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business

strategy of that organization.

Adherence to the STLC contributes to the effectiveness of testing process by :

- Providing a systematic and structured approach to testing, ensuring thorough coverage and of the software under test.
- Identifying defects early in the development lifecycle, reducing the cost and effort of fixing them.
- Facilitating better coordination and communication among project stakeholders by clearly defining test objectives, deliverables and timelines.

2 Provide an overview of different types of software testing, including functional testing, non-functional testing, manual testing, and automated testing. Discuss the purpose and methodologies associated with each type.

Testing is a process of analyzing software item to detect the difference between existing and required conditions and to evaluate features of software item.

### 1. Manual Testing:

Manual Testing involves human testers executing test cases manually without the use of automated tools. It is useful for exploratory testing and those testing scenarios where automation is not feasible.

Methodologies :

Ad-hoc testing : Informal testing performed without predefined test cases, often used to uncover unexpected defects.

Exploratory testing : simultaneously designing and executing test cases to explore the softwares functionality and uncover defects.

Regression testing : Repeating test cases to ensure that previously developed and tested software still performs correctly after changes have been made.

## 2. Automated Testing

Automated testing involves using software tools to execute test cases automatically thereby increasing efficiency, repeatability and coverage.

Methodologies :

Test scripting : Writing test scripts to automate the execution of test cases.

Test Framework : Utilizing testing framework such as selenium, JUnit, TestNG to organise and execute automated tests.

Load Testing : Using automated tools to stimulate high volume of user traffic to access systems performance under load conditions.

### 3. Functional Testing :

Functional testing is testing focuses on verifying that the software functions correctly according to the specified requirements. It ensures that the system performs its intended functions accurately.

#### Methodologies :

Unit Testing : Testing individual units or components of the software in isolation.

Integration Testing : Testing the interactions between different units or components to ensure they work together.

System Testing : Testing the entire system as a whole to verify that it meets the specified requirements.

User Acceptance Testing : Testing performance by end user to validate whether the system meets requirements.

### 4. <sup>Non</sup>Functional Testing :

Non-Functional Testing focuses on evaluating aspects of the software other than its specific functionalities. It ensures software meets performance, security, usability.

#### Methodologies :

Performance - Testing : Evaluating the performance characteristics of system, such as response time, scalability.

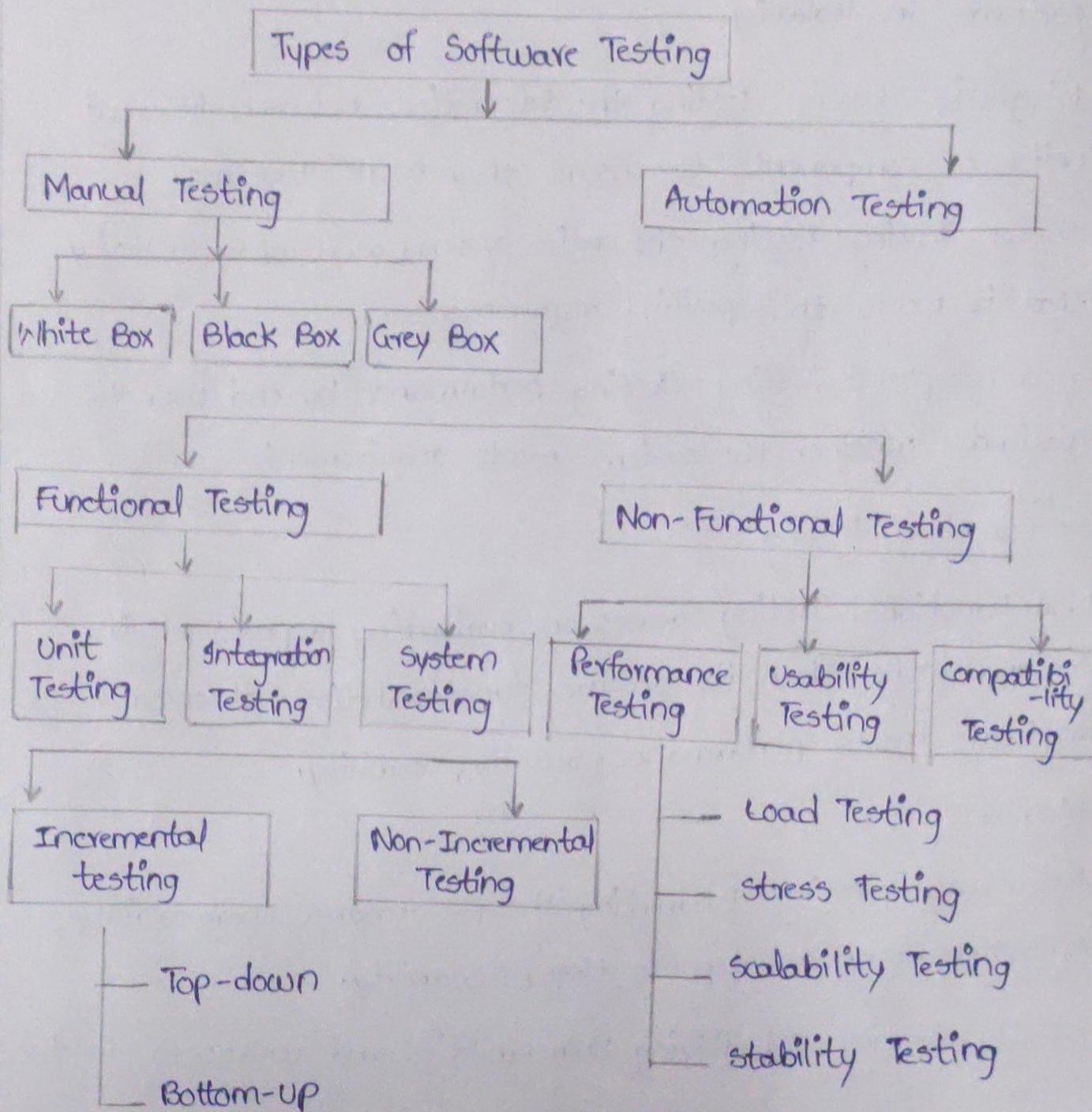
Security - Testing : Identifying vulnerabilities and weakness in the

system to ensure data protection.

Usability Testing : Assessing the user friendliness and intuitiveness of software interface.

Compatibility Testing : Ensuring that the software functions correctly across different platforms, browsers, devices.

Reliability Testing : Testing the reliability and stability of the software under various conditions.



3 Explain the concept of functional testing, focusing on its black box testing approach. Discuss the strategies and methodologies used in black box testing to verify the functionality of a software application. How does black box testing accommodate for different perspectives and requirements of stakeholders?

Functional testing in its black box approach, focuses on evaluating the functionality of a software application without examining its internal code structure. In black box testing, testers are only connected with inputs to square software and outputs it generates, treating the software as "black box".

Strategies and Methodologies in Black Box Testing :

Equivalence Partitioning :

This strategy divides input domain as valid and invalid inputs into groups based on similarities. This ensures all scenarios within a group are tested efficiently.

Boundary Value Analysis :

Focuses on testing inputs at the edges of acceptable ranges. Like testing a text field with a 1 character and 100 character input.

Error Guessing :

Leveraging experience, testers anticipate potential errors and design test cases to trigger them.

### State Transition Testing :

Focuses on how the application behaves when transitioning between different states. This ensures smooth operation during these transitions.

### Use case Testing :

Derives test cases based on how end users will interact with the application.

### Accomodating Stakeholders:

Black box testing addresses stakeholders needs through its focus on requirements.

### Requirements as Source:

Test cases stem from the documented requirements of the applications. This ensures the software fulfills its intended functionalities as envisioned by stakeholders.

### Diverse Perspective:

Black box testing doesn't require programming knowledge allowing testers from various backgrounds to participate. This incorporates the perspective of different stakeholders.

### Focus on Functionality:

By slowly examining functionalities, black box testing ensures the application delivers the features stakeholders needs.

### User-Centric Testing:

By treating the software as a black box and focusing on

inputs and outputs, emphasizes end user experience.

- 4 Describe the concept of the static Testing strategies in software development, focusing on formal technical reviews (peer reviews), walkthroughs, and other static analysis techniques. Discuss the benefits of incorporating these strategies into the software development lifecycle and explain how they contribute to early defect detection and prevention. Static Testing techniques analyze software artifacts (documents, code) without actual execution. This proactive approach identifies defects early in the development life cycle.

Formal Technical Reviews :

- A group setting where developers systematically examine each other work.
- Focuses on identifying logic flaws, coding inconsistency potential security , vulnerabilities.
- Improved quality of code , knowledge sharing, early defect detection and fostering culture.

Walkthrough :

- A developer leads a team through the code, and explaining design decisions and implementations.
- Emphasizes on clarity, logic flow, potential errors, adherence to requirements.
- Early requirements or identification of design flaws, promotes

code maintainability.

Static Code Analysis Tools :

- Automated tools that can scan code for various issues like syntax errors, security, vulnerabilities, code smells, adherence to code standards.
- Improve code quality enforces coding standards.

Other static Analysis Techniques :

Data Flow Analysis : Examines how data flow through code, identifying potential issues.

Control Flow Analysis : Reviews how the program flow might behave based on different input scenarios helping uncover potential logic errors.

Model-Based Testing : Creates a model of expected system behavior and compares it to the code, highlighting inconsistency.

Benefits of static Testing strategies :

Early Defect Detection :

Identifies issues before code execution, when fixing them is easier and cheaper.

Improved code Quality :

Ensures adherence to coding standards, reduces the complexity and promotes maintainable code.

Reduced Development Costs:

Fixing defects early avoids costly rework and delays later in the development cycle.

Enhanced Security:

Helps identify potential security vulnerabilities early on making the software more secure.

Knowledge sharing:

Fosters collaboration and knowledge transfer within development teams through reviews and walkthroughs.

- 5 outline the key components of test planning, management, execution, and reporting in the software testing process. Discuss the importance of thorough test planning in defining test objectives, scope, resources and schedules.

The software testing process involves several stages to ensure a software application functions as intended.

### Test Planning

Foundation of Successful Testing:

A well defined test plan lays the ground work for entire testing process, outlines the testing strategy, objectives, scope.

Defining objectives & scope:

The plan clearly defines what will be tested and what won't. This ensures focused testing efforts.

### Resource allocation:

The plan identifies the testing tools, personnel, and infrastructure needed to execute the tests effectively.

### Schedule creation:

A realistic timeline is established for each testing phase ensuring timely completion.

### Test Management

#### Overseeing the testing process:

Test Management involves overseeing the execution of the test plan, monitoring progress, managing resources.

### Risk Management:

Potential testing risks are identified, and mitigation strategies are developed to ensure smooth execution.

### Communication & Collaboration:

Effective communication channels are established between testers, developers, and other stakeholders to ensure everyone is informed of testing progress.

### Test Execution

#### Putting the plan into action:

The test cases designed during the planning phase are executed against the software application.

## Data Preparation :

The test data that reflects real-world scenarios is created to thoroughly evaluate the software functionality under various conditions.

## Defect Logging :

Identified issues are documented with detailed descriptions, steps to reproduce.

## Test Reporting

### Communication Test Results :

Comprehensive test report summarizes the testing process, including the number of test cases executed identified defects.

### Analysis & Recommendations :

The report analyzes the testing result, identifies trends, and provides recommendations for improvement, like further testing or code modifications.

### Stakeholder Communication :

Test reports are effectively communicated to relevant stakeholders to keep everyone informed of testing process.

## Importance of thorough Test planning

### Clarity & Focus :

A well defined plan provides clarity and focus for the entire testing report. Testers understand their objectives, what needs to be tested, and resources available.

### Optimized Resource Allocation:

Planning helps in efficiently allocating resources to ensure all aspects of software are tested.

### Risk Mitigation:

Proactive identification of potential risks allow for the development of contingency plans, minimizing disruptions.

### Improved Communication:

A clear plan facilitates better communication between testers, developers and stakeholders, leading to a more collaborative and efficient testing process.