

SQA Fundamentals

1

Chapter 17, 18

Contents

2

- Testing Definition
- Types of Testing
- Principles of Testing
- V-Model
- STLC

Testing Definition

3

- Software Testing is a process used to identify the **correctness, completeness and quality** developed software.
- It includes a set of activities conducted with the intent of **finding errors in software** so that it could be corrected before released to the end users.
- Ensure that the software system is ***Defect*** free (actual results match the expected results)
- Helps to identify **errors, gaps or missing requirements** in contrary to the actual requirements

Testing Definition Cont...

4

- Sometimes - Software testing can be defined as a **White Box** and **Black Box Testing**.
- Software Testing means Verification of Application Under Test (AUT)
- In simple Word, software testing is an activity to check that the software is **defect free**.

Types of Testing

5

- Typically Testing is classified into three categories
 - Functional Testing
 - Non-Functional Testing or Performance Testing
 - Maintenance (Regression and Maintenance)

Functional Testing

6

- FUNCTIONAL TESTING is a type of software testing whereby the system is tested against the ***functional requirements/specifications***.
- This testing involves checking of **User Interface, APIs, Database, security, client/ server applications and *functionality*** of the AUT.
- The testing can be done either manually or using automation.

Why Functional Testing?

7

- **Mainline functions:** Testing the main functions of an application
- **Basic Usability:** It involves basic usability testing of the system. It checks whether a user can freely navigate through the screens without any difficulties.
- **Accessibility:** Checks the accessibility of the system for the user
- **Error Conditions:** Usage of testing techniques to check for error conditions. It checks whether suitable error messages are displayed.

Steps of FT

8

- Identify functions that the software is expected to perform.
- Create input data based on the function's specifications.
- Compute the expected outcomes with the selected test input values
- Execute the **test case**
- Compare the actual and expected outputs

Functional Testing

9

- **Unit Testing**
- **Integration Testing**
- Smoke
- Sanity Testing
- **UAT (User Acceptance Testing)**
- **White Box**
- **Black Box**
- **Regression**
- Localization
- Globalization
- Interoperability

Tools

10

- **Ranorex Studio** — all-in-one functional test automation for desktop, web, and mobile apps with built-in Selenium WebDriver.
- **Selenium**- Popular Open Source Functional Testing Tool
- **QTP**- Very user-friendly Functional Test tool by HP
- **JUnit**- Used mainly for Java applications and this can be used in Unit and System Testing
- **soapUI**- This is an open source functional testing tool, mainly used for Web service testing. It supports multiple protocols such as HTTP, SOAP, and JDBC.
- **Watir** - This is a functional testing tool for web applications. It supports tests executed at the web browser and uses a ruby scripting language

Non Functional Testing

11

- Non-functional testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application
- Check how many people can simultaneously login into a software.

Objectives of NFT

12

- Non-functional testing should increase usability, efficiency, maintainability, and portability of the product.
- Helps to reduce production risk and cost associated with non-functional aspects of the product.
- Optimize the way product is installed, setup, executes, managed and monitored.
- Collect and produce measurements, and metrics for internal research and development.
- Improve and enhance knowledge of the product behavior and technologies in use.

Types of NFT

13

- **Performance Testing**
- **Load Testing**
- Volume Testing
- **Stress Testing**
- **Security Testing**
- Installation Testing
- Penetration Testing
- Compatibility Testing
- Migration Testing

Functional Testing	Non-Functional Testing
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first	Non-functional testing should be performed after functional testing
Manual Testing or automation tools can be used for functional testing	Using tools will be effective for this testing
Business requirements are the inputs to functional testing	Performance parameters like speed, scalability are inputs to non-functional testing.
Functional testing describes what the product does	Nonfunctional testing describes how good the product works
Easy to do Manual Testing	Tough to do Manual Testing

Maintenance Testing

15

- Regression
 - Maintenance
-
- More than 150 types of testing types and still adding

Principles of Software Testing

16

- Achieve optimum test results while conducting software testing (Without deviating from goal)
 - How will you determine that you are following the right strategy for testing?
 - You need to stick to some basic testing principles.
-
- 1) Exhaustive testing is not possible
 - 2) Defect Clustering
 - 3) Pesticide Paradox
 - 4) Testing shows a presence of defects
 - 5) Absence of Error - fallacy
 - 6) Early Testing
 - 7) Testing is context dependent

1) Exhaustive testing is not possible

17

- Complete testing is not possible
- Optimal amount of testing based on the risk assessment of the application

Suppose you have 15 input fields to test, each having 5 possible values, the number of combinations to be tested would be 5^{15}

2) Defect Clustering

18

- Defect Clustering which states that a **small number of modules** contain most of the defects detected
- **Pareto Principle to software testing:** approximately 80% of the problems are found in 20% of the modules.
- The modules may be complex, coding related to such modules may be complicated
- If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

3) Pesticide Paradox

19

- Problems
 - Same set of repetitive tests are unable to discover new defect
- Solutions
 - Test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.
 - Improve the existing methods to make testing more effective.
- Even after all this **sweat & hard work** in testing, you can never claim your **product is bug-free**.

4) Testing shows a presence of defects

20

- Testing talks about the **presence of defects** and **don't talk** about the absence of defects
- ST reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.
- Software product 99% bug-free after all testing but it does not meet the needs & requirements of the clients ???

5) Absence of Error - fallacy

21

- It is possible that software which is 99% bug-free is still unusable
- The system is tested thoroughly for the wrong requirement.
- Software testing is not mere finding defects, but also to check that software addresses the business needs.
- Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.

6) Early Testing

22

- Testing should start as early as possible in the Software Development Life Cycle
- Any defects in the requirements or design phase are captured in early stages
- **But how early one should start testing?**
- Recommended that you start finding the bug the moment the requirements are defined

7) Testing is context dependent

23

- Testing is context dependent
- Different strategy should follow for different applications
- Use a different approach, methodologies, techniques, and types of testing depending upon the application type
- **Example:** Testing strategy of E-commerce site will be same as the testing of an ATM machine

V Model in ST

24

- SDLC vs STLC
- Waterfall Model vs V Model (extension of Waterfall)
- There is a corresponding testing phase for each software development phase.
- Testing in V-model is done in parallel to SDLC stage.
- Testing is done as a subproject of SDLC.

Waterfall Model

25

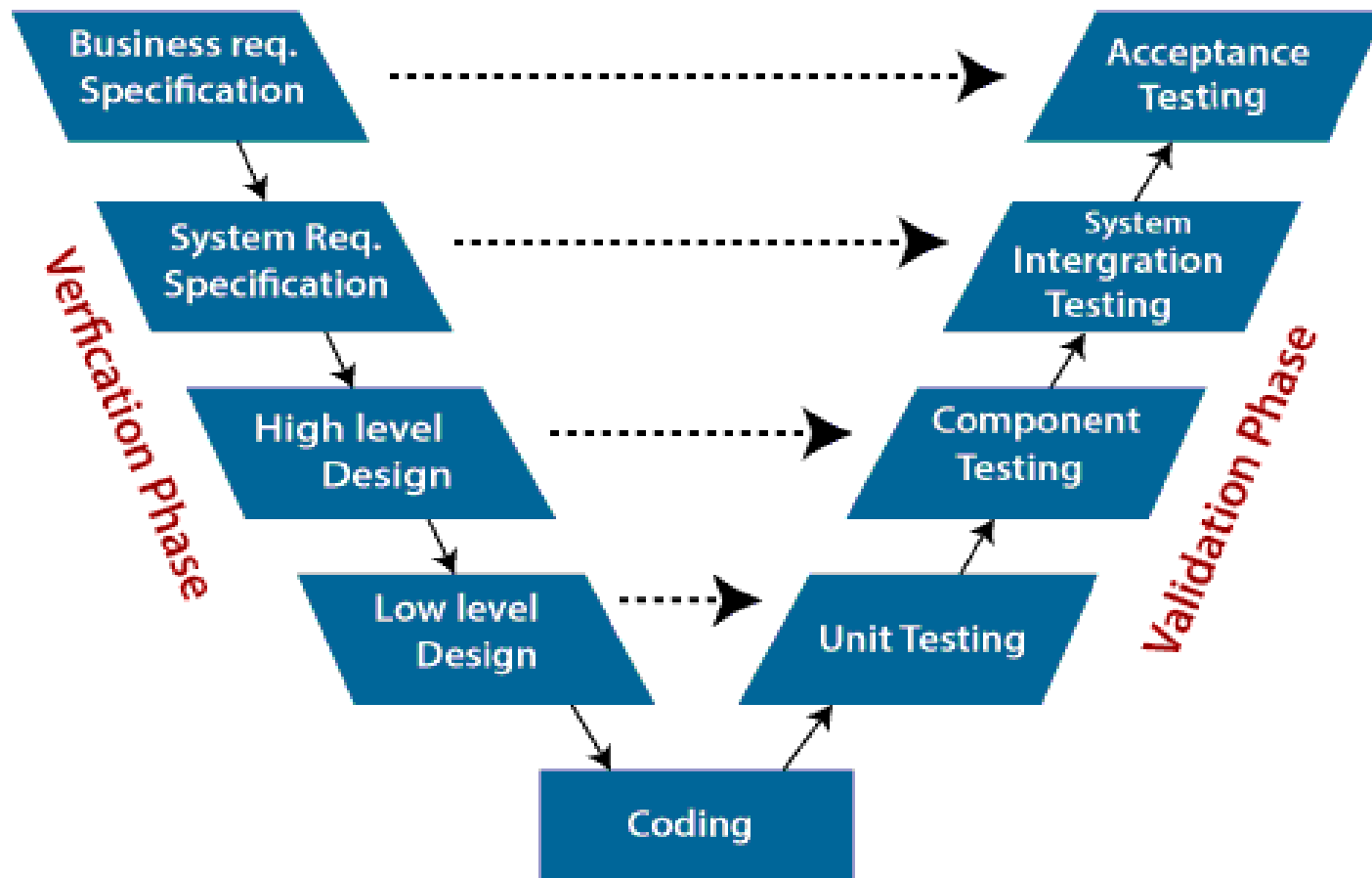
- Requirements gathering
- Design
- Develop
- Test
- Maintenance

V-Model

V- Model

Developer's life Cycle

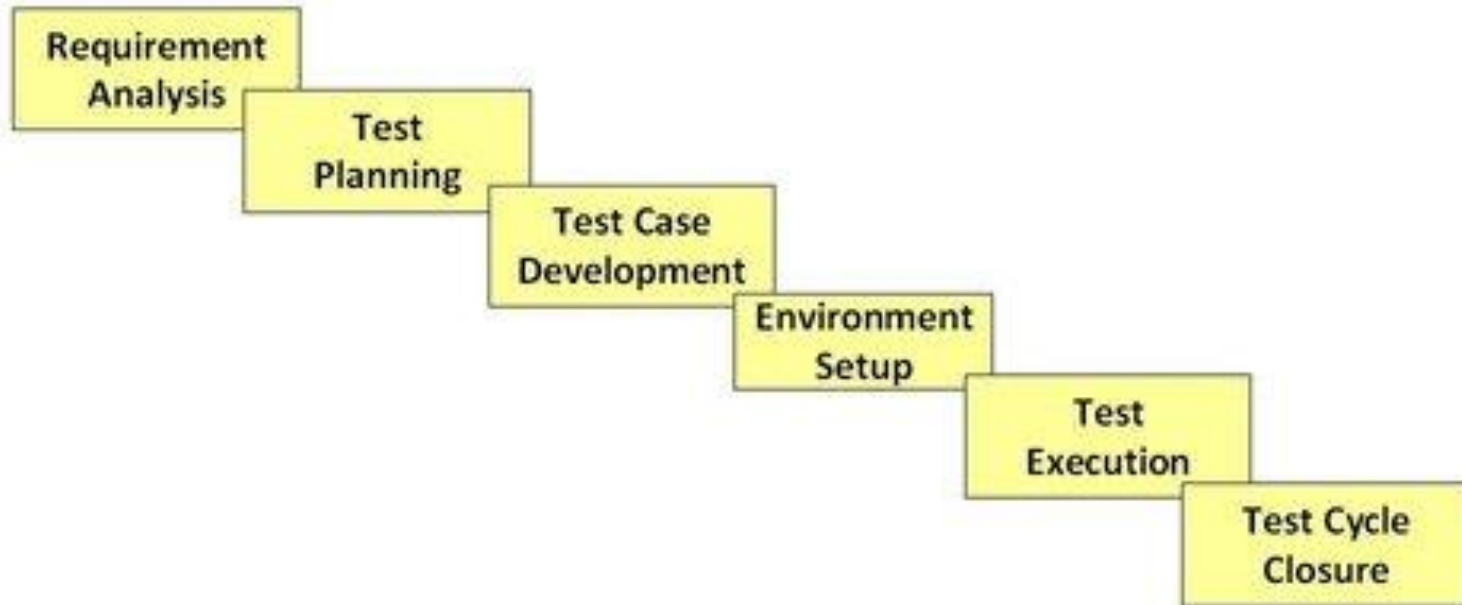
Tester's Life Cycle



STLC

27

- Software Testing Life Cycle



Requirements Analysis

28

- Identify types of tests to be performed.
- Gather details about testing priorities and focus
- Prepare Requirement Traceability Matrix (**RTM**)
- Identify test environment details where testing is supposed to be carried out
- Automation feasibility analysis (if required)

Test Planning

29

- Preparation of test plan/strategy document for various types of testing
- Test tool selection
- Test effort estimation
- Resource planning and determining roles and responsibilities.
- Training requirement

Test Case Development

30

- Create test cases, automation scripts (if applicable)
- Review and baseline test cases and scripts
- Create test data (If Test Environment is available)

Test Environment Setup

31

- Understand the required architecture, environment set-up and prepare hardware and software requirement list for the Test Environment.
- Setup test Environment and test data
- Perform smoke test on the build

Test Execution

32

- Execute tests as per plan
- Document test results, and log defects for failed cases
- Map defects to test cases in RTM
- Retest the Defect fixes
- Track the defects to closure

Test Cycle Closure

33

- Evaluate cycle completion criteria based on Time, Test coverage, Cost, Critical Business Objectives, Quality.
- Prepare test metrics based on the above parameters.
- Document the learning out of the project
- Prepare Test closure report
- Qualitative and quantitative reporting of quality of the work product to the customer.
- Test result analysis to find out the defect distribution by type and severity.

Terminologies (Need to know)

34

- Different Types of Testing
- Testing Techniques
- Tools-Selenium
- Quality
- RTM
- Test Case, Test Coverage
- Defect, Error, Bug

Thank You

35