

## Unit 4: Software Engineering

### Software Quality Concepts:

- It is a way to assure quality in the software.
- It is the set of activities which ensure processes, procedures as well as standards suitable for the project and implemented correctly.
- Software Quality Assurance is a process which works parallel to development of a software.
- It focuses on improving the process of development of software so that problems can be prevented before they become a major issue.

### Major Software Quality Assurance Activities

#### **SQA Management Plan:**

Make a plan how you will carry out the SQA throughout the project. Think which set of software engineering activities are the best for project.

Check level of SQA team skills.

#### **Set the Check Points:**

SQA team should set checkpoints.

Evaluate the performance of the project on the basis of collected data on different check points.

#### **Multi testing Strategy:**

Do not depend on single testing approach.

When you have lot of testing approaches available use them.

#### **Measure Change Impact:**

The changes for making the correction of an error sometimes re-introduces more errors keep the measure of impact of change on project.

#### **Manage Good Relations:**

In the working environment managing the good relation with other teams involved in the project development is mandatory.

### Benefits of Software Quality Assurance (SQA)

- SQA produce high quality software.
- High quality application saves time and cost.

- SQA is beneficial for better reliability.
- SQA is beneficial in the condition of no maintenance for long time.
- High quality commercial software increase market share of company.
- Improving the process of creating software.
- Improves the quality of the software

### Software Reliability

A simple measure of reliability is *mean-time-between-failure* (MTBF), where

$$MTBF = MTTF + MTTR$$

The acronyms MTTF and MTTR are *mean-time-to-failure* and *mean-time-to-repair*, respectively.

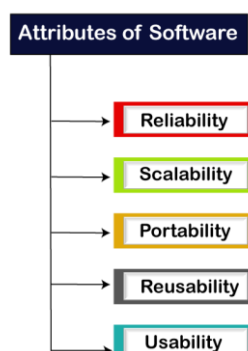
*Software availability* is the probability that a program is operating according to requirements at a given point in time and is defined as

$$\text{Availability} = [MTTF / (MTTF + MTTR)] \times 100\%$$

### Testing

Software testing is widely used technology because it is compulsory to test each and every software before deployment.

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.



## Testing Strategy

### Unit testing

Unit testing starts at the centre and each unit is implemented in source code.

### Integration testing

An integration testing focuses on the construction and design of the software.

### Validation testing

Check all the requirements like functional, behavioural and performance requirement are validated against the construction software.

### System testing

System testing confirms all system elements and performance are tested entirely.

### White-box testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing.

In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

- **Control-flow testing** - The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
- **Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

### Black-box testing

It is carried out to test functionality of the program. It is also called 'Behavioural' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

## Types of Black Box Testing

There are many types of Black Box Testing but the following are the prominent ones:

- **Functional testing** - This black box testing type is related to the functional requirements of a system. It is done by software testers.
- **Non-functional testing** - This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing** - Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

## What is a Test Case?

The test case is an in-details document that includes **all possible inputs** such as **positive and negative**, and **the navigation steps**. These are implemented during the testing process to check whether the software application is performing the task for that it was developed or not.

Differences between Test Case and Test Scenarios:

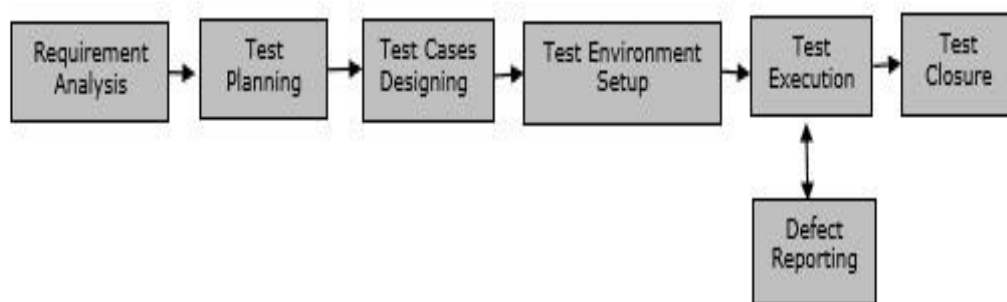
1.	The test case is a detailed document, which provides information about the <b>testing strategy, testing process, preconditions, and expected output</b> .	The test scenarios are those derived from the <b>use case</b> and give <b>the one-line information about what to test</b> .
2.	It includes all the <b>positive and negative inputs, navigation steps, Expected results, pre and post condition, etc.</b>	Test scenarios are one-liner statement, but it is connected with several test cases.
3.	These are <b>low-level actions</b> .	These are <b>high-level actions</b> .

4.	It takes <b>more time</b> as compared to test scenarios.	It takes <b>less time</b> as compared to test cases.
5.	The test case is work on the basics of " <b>How to be tested</b> ".	The test scenarios are work on the basic to " <b>What to be tested</b> ".

## STLC

- STLC ---→ Software Testing Life Cycle.
- STLC is a sequence of different activities performed by the testing team to ensure the quality of the software or the product.
- STLC is an integral part of Software Development Life Cycle (SDLC).

## STLC PHASES



- **Requirement Analysis** – When the SRD is ready and shared with the stakeholders, the testing team starts high level analysis concerning the AUT (Application under Test).
- **Test Planning** – Test Team plans the strategy and approach.
- **Test Case Designing** – Develop the test cases based on scope and criteria.
- **Test Environment Setup** – When integrated environment is ready to validate the product.
- **Test Execution** – Real-time validation of product and finding bugs.

- **Test Closure** – Once testing is completed reports and results are documented.

### **Verification and Validation**

Verification and Validation is the process of investigating that a software system satisfies specifications and standards and it fulfills the required purpose.

**1) Verification:** *Are we building the product right?*

**2) Validation:** *Are we building the right product?*

#### **Verification**

Verification is the process of checking that a software achieves its goal without any bugs.

It is the process to ensure whether the product that is developed is right or not.

It verifies whether the developed product fulfills the requirements that we have.

Verification is **Static Testing**.

#### **Validation**

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements.

- It is the process of checking the validation of product
- It checks what we are developing is the right product.
- It is validation of actual and expected product.

Validation is the **Dynamic Testing**.