

**COURSE TITLE:** **SYSTEM ANALYSIS AND DESIGN**

**PAPER CODE:** CSE 4205

**LECTURER:** **MASESE CHUMA- 0701260004**

**TOPIC:** **TESTING AND QUALITY ASSURANCE**

---

### **Objective**

By the end of this session, students will be able to:

To discuss system **testing and quality assurance**

### **TESTING AND QUALITY ASSURANCE**

The software system needs to be checked for its intended behavior and direction of progress at each development stage to avoid duplication of efforts, time and cost overruns, and to assure completion of the system within stipulated time.

System testing and quality assurance come to aid for checking the system. It includes:

- Product level quality (Testing)
- Process level quality.

Let us go through them briefly:

#### **Testing**

Testing is the process or activity that checks the functionality and correctness of software according to specified user requirements in order to improve the quality and reliability of system. It is an expensive, time consuming, and critical approach in system development which requires proper planning of overall testing process.

A successful test is one that finds the errors. It executes the program with explicit intention of finding error, i.e., making the program fail. It is a process of evaluating system with an intention of creating a strong system and mainly focuses on the weak areas of the system or software.

#### **Characteristics of System Testing**

System testing begins at the module level and proceeds towards the integration of the entire software system. Different testing techniques are used at different times while testing the system. It is conducted by the developer for small projects and by independent testing groups for large projects

#### **Stages of System Testing**

The following stages are involved in testing:

#### **Test Strategy**

It is a statement that provides information about the various levels, methods, tools, and techniques used for testing the system. It should satisfy all the needs of an organization.

### **Test Plan**

It provides a plan for testing the system and verifies that the system under testing fulfills all the design and functional specifications. The test plan provides the following information:

#### **Objectives of each test phase**

1. Approaches and tools used for testing
2. Responsibilities and time required for each testing activity
3. Availability of tools, facilities, and test libraries
4. Procedures and standards required for planning and conducting the tests
5. Factors responsible for successful completion of testing process

### **Test Case Design**

- Test cases are used to uncover as many errors as possible in the system.
- A number of test cases are identified for each module of the system to be tested.
- Each test case will specify how the implementation of a particular requirement or design decision is to be tested and the criteria for the success of the test.
- The test cases along with the test plan are documented as a part of a system specification document or in a separate document called **test specification** or **test description**.

### **Test Procedures**

It consists of the steps that should be followed to execute each of the test cases. These procedures are specified in a separate document called test procedure specification.

This document also specifies any special requirements and formats for reporting the result of testing.

### **Test Result Documentation**

Test result file contains brief information about the total number of test cases executed, the number of errors, and nature of errors. These results are then assessed against criteria in the test specification to determine the overall outcome of the test.

### **Types of Testing**

Testing can be of various types and different types of tests are conducted depending on the kind of bugs one seeks to discover:

#### **Unit Testing**

Also known as Program Testing, it is a type of testing where the analyst tests or focuses on each program or module independently. It is carried out with the intention of executing each statement of the module at least once.

- In unit testing, accuracy of program cannot be assured and it is difficult to conduct testing of various input combination in detail.
- It identifies maximum errors in a program as compared to other testing techniques.

### **Integration Testing**

In Integration Testing, the analyst tests multiple modules working together. It is used to find discrepancies between the system and its original objective, current specifications, and systems documentation.

- Here the analysts are trying to find areas where modules have been designed with different specifications for data length, type, and data element name.
- It verifies that file sizes are adequate and that indices have been built properly.

### **Functional Testing**

Function testing determines whether the system is functioning correctly according to its specifications and relevant standards documentation. Functional testing typically starts with the implementation of the system, which is very critical for the success of the system.

Functional testing is divided into two categories:

- **Positive Functional Testing:** It involves testing the system with valid inputs to verify that the outputs produced are correct.
- **Negative Functional Testing:** It involves testing the software with invalid inputs and undesired operating conditions.

### **Rules for System Testing**

- To carry out system testing successfully, you need to follow the given rules:
- Testing should be based on the requirements of user.
- Before writing testing scripts, understand the business logic should be understood thoroughly.
- Test plan should be done as soon as possible.
- Testing should be done by the third party.
- It should be performed on static software.
- Testing should be done for valid and invalid input conditions.
- Testing should be reviewed and examined to reduce the costs.
- Both static and dynamic testing should be conducted on the software.
- Documentation of test cases and test results should be done.

## **QUALITY ASSURANCE**

It is the review of system or software products and its documentation for assurance that system meets the requirements and specifications.

- Purpose of QA is to provide confidence to the customers by constant delivery of product according to specification.
- Software quality Assurance (SQA) is a technique that includes procedures and tools applied by the software professionals to ensure that software meet the specified standard for its intended use and performance.
- The main aim of SQA is to provide proper and accurate visibility of software project and its developed product to the administration.
- It reviews and audits the software product and its activities throughout the life cycle of system development.

### **Objectives of Quality Assurance**

The objectives of conducting quality assurance are as follows:

- To monitor the software development process and the final software developed.
- To ensure whether the software project is implementing the standards and procedures set by the management.
- To notify groups and individuals about the SQA activities and results of these activities.
- To ensure that the issues, which are not solved within the software are addressed by the upper management.
- To identify deficiencies in the product, process, or the standards, and fix them.

### **Levels of Quality Assurance**

There are several levels of QA and testing that need to be performed in order to certify a software product.

#### **Level 1: Code Walk-through**

At this level, offline software is examined or checked for any violations of the official coding rules. In general, the emphasis is placed on examination of the documentation and level of in-code comments.

#### **Level 2: Compilation and Linking**

At this level, it is checked that the software can compile and link all official platforms and operating systems.

#### **Level 3: Routine Running**

At this level, it is checked that the software can run properly under a variety of conditions such as certain number of events and small and large event sizes etc.

#### **Level 4: Performance test**

At this final level, it is checked that the performance of the software satisfies the previously specified performance level.