order to know their needs and expectations from the product. The answers for the following questions should be ready—

- Who will use the product?
- What the product is used for?
- How the product will work?
- What software/hardware the product uses?

The following approach given in Figure 12.3 can be used to analyze the product.

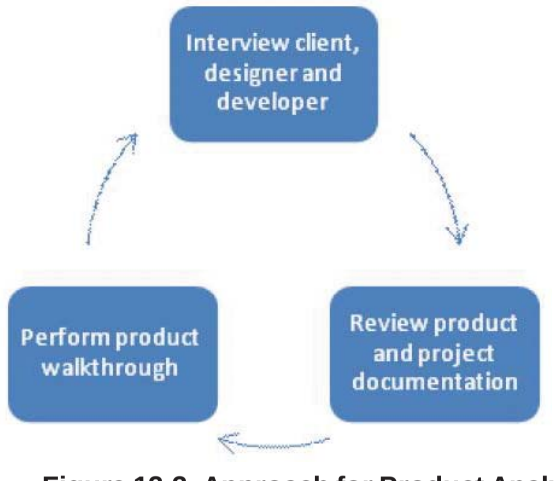


Figure 12.3: Approach for Product Analysis

Develop test strategy

It is a very critical step in making a test plan. A test strategy document is a high level document. It is usually developed by Test Manager. This document defines the objectives of the project's testing and how it will be achieved. It determines the testing efforts and costs. A test strategy involves the following steps shown in Figure 12.4.



Figure 12.4: Process of Test Strategy

involved with database design. Data can be viewed in terms of physical and logical view. Physical view, logical view, schemas and sub schemas were discussed. We learned about the various types of relationships. Data structure plays an important role in the mathematical organization of data inside a system. We discussed about three types of data structures, namely, hierarchical, network and relational data structure. Role of entity and attributes were seen properly. Database administrator plays a vital role in database design. The role of database administrator was discussed. Finally, we have discussed how to apply normalization.

In this unit, we will learn about system testing and its process. We will see why system testing is so important. The nature of test data will be discussed. We will see the importance of test plan, components of a test plan, types of test plan and creation of a test plan. Discussion will be done on activity network for system testing. Finally, the types of system test will be discussed. In the next unit, we will discuss about the quality assurance of a system.

12.3 NEED FOR SYSTEM TESTING

System testing is a level of testing that helps to validate a complete and fully-integrated software product. System testing is very important in the process of development of a software product. Its purpose is to evaluate the end-to-end system specifications. System testing is viewed as a series of different tests whose main goal is to exercise the full computer-based system.

System testing evaluates the compliance of the system with the corresponding system requirements. The output of the system testing is the observed behavior of individual components or the whole system when it is tested. System testing is done on the whole system in the context of either system requirement specifications or functional requirement specifications or both. It tests the design and behavior of the system so that it meets the expectations of the customers. It is performed to test the system beyond the requirements mentioned in the Software Requirements Specifications (SRS).

System testing is a level of testing performed by a testing team. It is independent of the development team and helps in testing the quality of the system impartial. It does both functional and non-functional testing. It is a black box testing method. Black box testing method does not require internal knowledge of the code. Input is provided to the system and the output is verified. The knowledge of the internal system is not required. System testing is performed after the integration testing step and before the acceptance testing step. Figure 12.1 presents an overview of levels of testing.

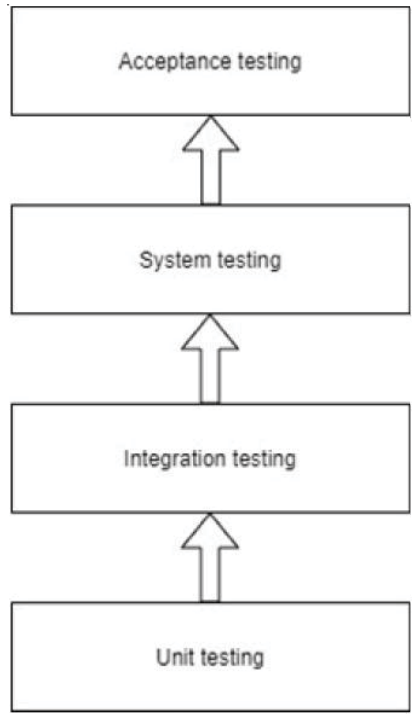


Figure 12.1: Levels of Testing

In brief, system testing is important due to the following reasons—

- It is the first level of software testing. It tests the software/application as a whole.
- It verifies and validates the business, technical, functional and non-functional requirements of the software.
- It verifies and validates the software application architecture.
- It is done on staging environment. It closely resembles the production environment where the final software will be developed.

- **Defect management**: It gives details about the bugs. It specifies how bugs will be reported; to whom and what each bug report needs to be accompanied by. For example, on occurrence of a bug, should it be reported with text logs, screenshots or videos of the code where it has occurred?
- **Risk management**: It gives details about risks that may occur during software testing and what risks the software may suffer from if sufficient testing is not done before the release of the software.
- **Exit parameters**: It gives details of when to stop the testing activities. It gives the description of the results that are expected from the QA operations. It gives a benchmark to the testers so that they can compare with the actual results.

12.5.3 Types of Test Plan

There are three types of test plan—

- Master test plan
- Phase test plan
- Specific test plan

Master test plan: It is a test plan where multiple levels of testing are involved. It includes the complete test strategy.

Phase test plan: It is a test plan where only one phase of the testing strategy is involved. For example, a list of test cases, a list of tools, etc.

Specific test plan: It is a test plan that addresses specific or major types of testing like load testing, security testing, performance testing, etc. A specific test plan is usually designed for non-functional testing.

12.5.4 Creation of A Test Plan

The process of creating a test plan involves the following steps:

Product Analysis

It is very important to analyze a product thoroughly before testing it. The clients and the end users should be researched thoroughly in

this offers transparency into the quality analysis activities so that all stakeholders get the idea of how the software will be tested. Quality Assurance (QA) managers or leaders build the test plan based on inputs from QA and non-QA team members. For its creation, it should not take more than one-third of the time allocated for the entire project.

12.5.1 Importance of Test Plan

The test plan is important due to the following reasons:

- It helps individuals who are not part of the QA team like developers, business managers, etc. to understand how exactly the software will be tested.
- It offers a guide or blue print for QA engineers to conduct testing activities.
- The details of aspects such as test scope, estimation, strategy, etc. are clearly written. Providing all these information into a single document makes it easier to be reviewed by the management personnel or to be reused in other projects.

12.5.2 Components of A Test Plan

The components of a test plan are—

- **Scope**: The details about the objectives of a particular project are written. It gives the details of the user scenarios to be used in the test. If necessary, the scope can also specify the scenarios or issues that will not be covered in the project.
- **Schedule**: It gives details of the start dates and deadlines for testers so that results are delivered on time.
- **Resource allocation**: It gives details about which tester will work on which test.
- **Environment**: It provides details of the test environment which will include nature, configuration and availability.
- **Tools**: It gives details about the tools that will be used for testing, bug reporting and other relevant activities.

There are some criteria that must be satisfied in order to apply system testing. Theseinclude the following.

- Unit testing must be completed.
- Integration testing must be completed.
- The development of the whole software system should be complete.
- A software testing environment that closely resembles the production environment must be available. This is called staging environment.

12.3.1 Process of System Testing

System testing is a very important step in the process of software testing life cycle. (STLC). It is very important to define the process of system testing in order to ensure that its procedure is executed accurately. It helps to ensure that no critical step or detail is missed out. The steps involved in the process of system testing can vary from project to project or from organization to organization.

System testing process has the following steps—

- **Create a test plan**: Test plan is the initial step in the process of system testing. The leader or test manager defines the scope and objective of testing. It determines the strategies, makes the decision between manual and automated testing. It defines the entry and exit criteria, assigns roles and responsibilities, etc.
- **Test case creation**: It is the step from which the process of testing is initiated by the team. Test cases are prepared on the basis of use cases, testing requirements, user/client requirements such as user interface, technical requirements, functional and non-functional requirements, performance, etc.
- **Select test data**: Once the team develops the test cases, they work together in the process of selection and creation of required test data. Test data plays an important role in test execution. Test data are the inputs that help the team in getting expected results.

- **Test case execution**: The test cases that were created earlier will be executed by the team. The team constantly monitors the process and records any problems or issues or discrepancies encountered by the team during the process. The output of the testing is also recorded.
- **Bug reporting and fixation**: It is the step in which the team reports all the recorded bugs and issues if it exists to the concerned member of the team. Once the bugs and issues are reported, the programmer or the developer works with the testing team to fix and resolve the issues.
- **Repeat the test cycle (if required)**: Once all the issues and bugs are resolved and fixed, the team repeats the entire test cycle till the expected results are obtained.

The process of system testing is depicted in figure 12.2.

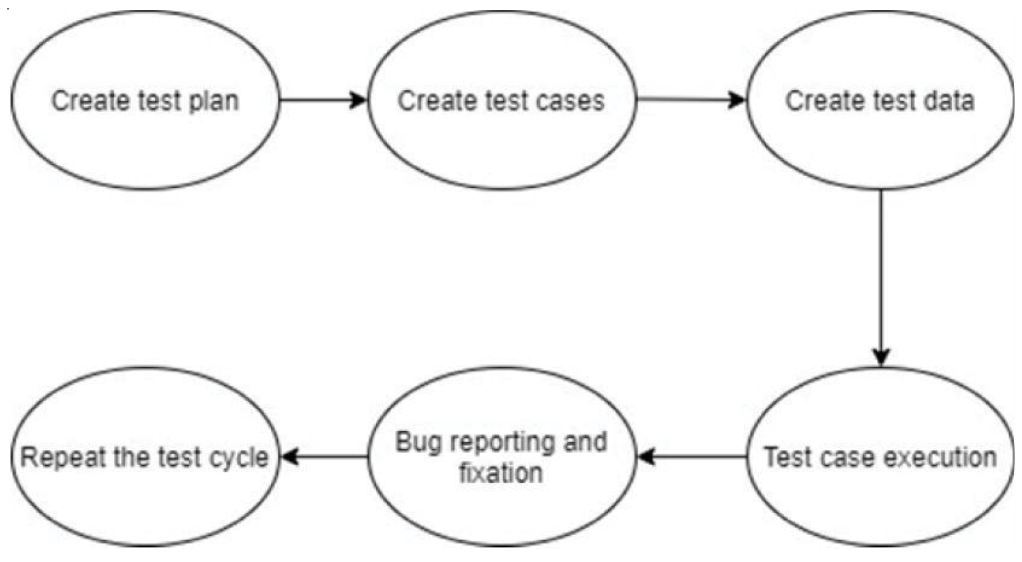


Figure 12.2: Process of System Testing

12.4 NATURE OF TEST DATA

The proper choice of test data is important. It is as important as the test itself. If the test data to be used as input are not valid or does not match the representative of the data to be provided by the user, then the reliability of the output is susceptible.

Test data may be artificially taken or taken from live data. It means test data can be created solely for test purpose or may be taken from user's actual files. Artificial data should be properly created. It should provide all combination of values or formats. It should be made possible to test all logic and transaction path subroutines. Live data are biased towards typical values. Artificial data provides extreme testing conditions. It provides extreme values for testing the boundaries of the candidate system.

For large and complex systems, dedicated computer programs are there to generate necessary test data. Substantial amount of time is saved by data generating programs both for the programmer and the test. In order to write an effective data-generating program, a familiarity with the system files and parameters is important.



CHECK YOUR PROGRESS

Q1. System testing follows white box testing. (State True/False).

Q2. System testing is performed before———————————— and after ————————————. (Fill up the blank)

Q3. How is test data taken?

12.5 TEST PLAN

A test plan is a detailed document that has a catalog containing the test strategy, objectives, schedule, estimations, deadlines and resources that are required for completing the particular project. It acts as a blueprint for running the tests in order to ensure that the software works properly. This process is controlled by the test managers. A good test plan should be dynamic in nature. It should be able to change according to progressions in the project and should stay current at all times. It is the point of reference based on which all the testing activities are executed and coordinated among a quality analysis team.

The test plan is shared with everyone associated with the project like business analyst, project managers, development team, etc. Doing