

Shubham P. Kumbhar

Wipro Assignment

Day-2

Que 1. What is the difference between software test planning and software test strategy?

Answer →

Test Plan	Test Strategy
A test plan can be defined as a document for a software project that defines the approach, scope, and intensity of the effort of software testing.	The test strategy is a set of instructions or protocols that explain the test design and determine how the test should be performed.
A test plan can be changed.	While the test strategy can't be changed.
The test plan happens independently.	While test strategy is often found as a part of a test plan.
The test plan describes the details.	While test strategy describes the general methodologies.
A test plan is done by the test administrator or test manager.	While The project manager does it.
A test plan is utilized at the project level.	While the test strategy is utilized at the association level.
A test plan has the essential objective of how to test when to test, and who will confirm it.	While test strategy has the essential objectives of what approach to pursue and which module to check.
Based on the testing strategies.	Based upon pre-defined standards.

Test Plan	Test Strategy
Different types of test plans are level-specific, type-specific, and master test plans.	Different types of test strategies are model-based, analytical, methodical, reactive, standard-compliant, consultative, and regression-averse strategy.
Only a single project is affected at a time.	Multiple projects can be impacted at a time.
It describes the general and common specifications in the testing of a specific object.	It describes the approaches in testing.

Que 2. What are different levels of software testing available?

Answer →

Testing levels are the procedure for finding the missing areas and avoiding overlapping and repetition between the development life cycle stages. We have already seen the various phases such as **Requirement collection, designing, coding testing, deployment, and maintenance** of [SDLC \(Software Development Life Cycle\)](#).

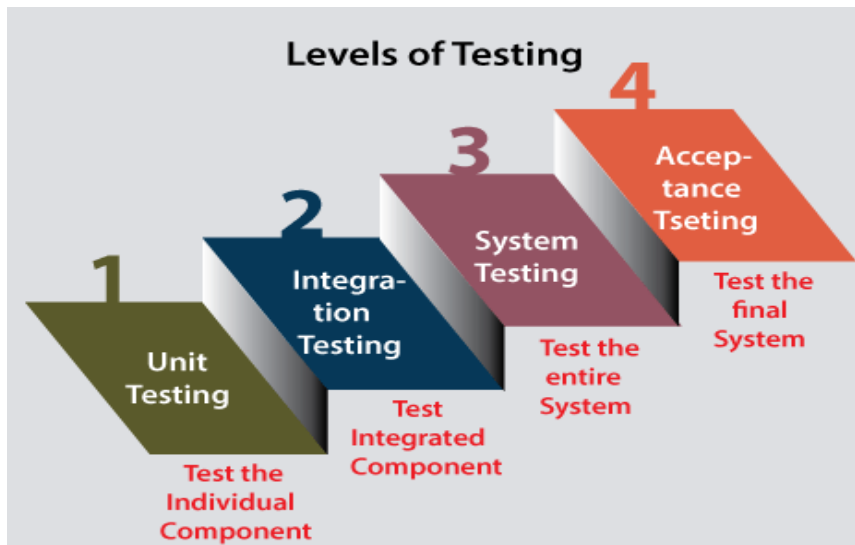
In order to test any application, we need to go through all the above phases of SDLC. Like SDLC, we have multiple levels of testing, which help us maintain the quality of the software.

Different Levels of Testing

The levels of software testing involve the different methodologies, which can be used while we are performing the software testing.

In [software testing](#), we have four different levels of testing, which are as discussed below:

1. **Unit Testing**
2. **Integration Testing**
3. **System Testing**
4. **Acceptance Testing**



As we can see in the above image that all of these testing levels have a specific objective which specifies the value to the software development lifecycle.

For our better understanding, let's see them one by one:

Level1: Unit Testing

Unit testing is the first level of software testing, which is used to test if software modules are satisfying the given requirement or not.

ADVERTISEMENT

The first level of testing involves **analysing each unit or an individual component** of the software application.

Unit testing is also the first level of [functional testing](#). The primary purpose of executing unit testing is to validate unit components with their performance.

A unit component is an individual function or regulation of the application, or we can say that it is the smallest testable part of the software. The reason of performing the unit testing is to test the correctness of inaccessible code.

Unit testing will help the test engineer and developers in order to understand the base of code that makes them able to change defect causing code quickly. The developers implement the unit.

Level2: Integration Testing

The second level of software testing is the **integration testing**. The integration testing process comes after **unit testing**.

It is mainly used to test the **data flow from one module or component to other modules**.

In integration testing, the **test engineer** tests the units or separate components or modules of the software in a group.

The primary purpose of executing the integration testing is to identify the defects at the interaction between integrated components or units.

When each component or module works separately, we need to check the data flow between the dependent modules, and this process is known as **integration testing**.

We only go for the integration testing when the functional testing has been completed successfully on each application module.

In simple words, we can say that **integration testing** aims to evaluate the accuracy of communication among all the modules.

Level3: System Testing

The third level of software testing is **system testing**, which is used to test the software's functional and non-functional requirements.

It is **end-to-end testing** where the testing environment is parallel to the production environment. In the third level of software testing, **we will test the application as a whole system**.

To check the end-to-end flow of an application or the software as a user is known as **System testing**.

In system testing, we will go through all the necessary modules of an application and test if the end features or the end business works fine, and test the product as a complete system.

In simple words, we can say that System testing is a sequence of different types of tests to implement and examine the entire working of an integrated software computer system against requirements.

Level4: Acceptance Testing

The **last and fourth level** of software testing is **acceptance testing**, which is used to evaluate whether a specification or the requirements are met as per its delivery.

The software has passed through three testing levels (**Unit Testing, Integration Testing, System Testing**). Some minor errors can still be identified when the end-user uses the system in the actual scenario.

In simple words, we can say that Acceptance testing is the **squeezing of all the testing processes that are previously done**.

The acceptance testing is also known as **User acceptance testing (UAT)** and is done by the customer before accepting the final product.

Usually, UAT is done by the domain expert (customer) for their satisfaction and checks whether the application is working according to given business scenarios and real-time scenarios.

Conclusion

A level of software testing is a process where every unit or component of a software or system is tested.

The main reason for implementing the **levels of testing** is to make the **software testing** process efficient and easy to find all possible test cases at a specific level.

To check the behavior or performance of software testing, we have various testing levels. The above-described software testing levels are developed to identify missing areas and understanding between the development life cycle conditions.

All these SDLC models' phases (**requirement gathering, analysis, design, coding or execution, testing, deployment, and maintenance**) undergo the process of software testing levels.