

Testing

What is integration testing?

This testing is the collection of the modules of the software, where the relationship and the interface between the different components are also tested. It needs coordination b/w the project level activities of integrating the constituent components at a time.

- * The purpose is to find out if these combined units interact without issues. By doing an integration test, testers can recognize errors b/w joined units.

- * Testers can also determine the reliability of the individual units. It tells if they perform expected functions. This test proceeds the unit test; as such, it is introduced on completion of unit testing.

→ The advantage of the test hierarchy is that integration testing can reveal anomalies missed during unit testing.

(OR)
Integration testing is when performing tests on multiple different modules and components together. The purpose of this testing is to examine how those various modules interact with each other.

Firstly, every module is tested separately to make sure it works smoothly and afterward the integration test is executed to check the performance of those modules collectively.

→ The integration testing is performed before the system testing and after unit testing.

→ This testing can be executed by QA testers, test engineers, and also developers.

- * There are few ways to perform integration testing like big-bang integration testing, top-down integration testing, bottom-up integration testing, ^{hybrid integration} etc.

- * Using a test plan could help significantly in the process of integration testing.

Advantages

- Helps to understand how different modules communicate with each other.
- Find bugs and defects in the interface between various components.
- can be performed by a wider variety of staff members.
- There are different approaches with different benefits to consider when performing integration testing like top-down, bottom-up, big bang integration.

System Testing

System testing is defined as the process of running tests on the entire software as a whole from a user point of view. The goal of this test is to check if the product is meeting user requirements and needs and finding bugs and defects that must be corrected in order to do so. System testing is also referred to as an end-to-end test because of the full coverage the test is providing and the advanced development stage the test is performing on.

- The system testing is executed mostly by test engineers and only after the integration testing is done.
- In system testing, there is no need for internal code knowledge because it includes black-box testing technique only.
- This particular testing can be categorized into functional testing (unit testing, regression testing, etc) and non-functional testing (performance testing, security testing and more).

Types

- functional
- scalability
- Reliability
- security
- performance testing
- usability
- installation testing etc,

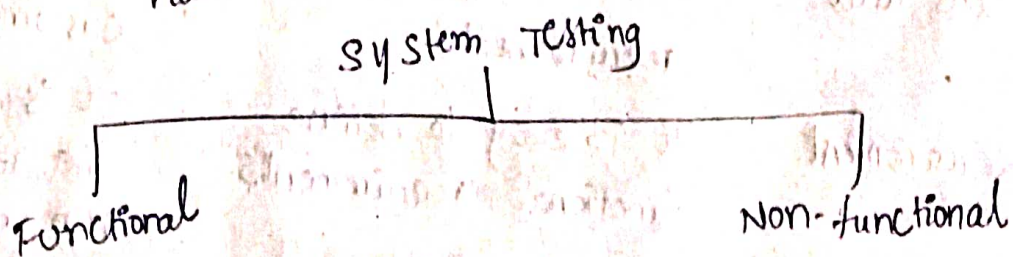
Advantages of System Testing :-

- End-to-End test that involves all the software components as a whole and checks for defects.
- Examine the software from a user point of view and simulate real-life scenarios.
- covers both functional and non-functional testing elements such as performance, usability, regression testing and more.
- No need for internal code knowledge.

How is system testing Done?

system testing is carried out as follows :

1. Test Plan :- create a test plan, including test cases and use cases.
2. Test data :- create test data and perform an automated test, followed by a manual test.
3. Test results :- Review the test result and write a report based on it. Log the errors and carry out regression testing after reporting the errors.
4. Debug :- Fix the errors found in the system.
5. Recycle :- Repeat the cycle and stop when all errors have been removed.



Functional Testing :-

- Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification.
- This testing mainly involves black-box testing, and it is not concerned about the source code of the application.
- Performed before non-functional testing.

Non-functional Testing :-

- It is a type of testing to check non-functional aspects (performance, usability, reliability etc.,) of a software application.
- It is explicitly designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing.

Parameters

Execution

Focus area

Requirement

Usage

Objective

Functional

It is performed before non-functional testing.

It is based on customer's requirements.

It is easy to define functional requirements.

Helps to validate the behaviour of the application

carried out to validate software actions.

Non-Functional

It is performed after the functional testing

It focuses on customer's expectation.

It is difficult to define the requirements for non-functional testing.

Helps to validate the performance of the application.

It is done to validate the performance of the software.

	<u>Function</u>	<u>Non-function</u>
Requirements	Functional testing is carried out using functional specification.	This Testing is carried out by performance specifications.
Manual Testing	Functional testing is easy to execute by manual testing.	It's very hard to perform non-functional testing manually.
Functionality	It describes what the product does.	It describes how the product works.
Example Test case.	check login Functionality	The dashboard should load in 2 seconds.

Black Box Testing and White Box Testing

Black Box Testing:- It is the method that does not consider the internal structure, design and product implementation to be tested. In other words, the tester does not know its internal functioning. The ~~black~~ it involves testing from an external or end-user perspective. Only evaluates the external behaviour of the system. The inputs received by the system and the outputs or responses it produces are tested. (high level testing)

→ Black box testing can be applied to virtually every level of software ~~engineering~~ ^{testing}: unit, integration, system and acceptance.

White Box Testing:-

It is the one that looks at the code and structure of the product to be tested and uses that knowledge to perform the tests. This method is used in the unit Testing phase, although it can also occur in the other stages such as Integration Testing.

→ For the execution of this method, the tester or the person who will use this method must have extensive knowledge of the technology used to develop the program.

Parameter

Def

only considers the external behaviour of the system; the internal workings of the software is not taken into account.

Alias

also known as data-driven, box testing, data and functional Testing.

Base of Testing

Testing is based on external expectations; internal behaviour of the application is unknown.

Usage

This type of testing is ideal for higher levels of testing like system, Acceptance testing.

Programming Knowledge

Not needed to perform

Objective

The main objective of this testing is to check what functionality of the system under test.

Tested by

end user, developer & Tester.

Time

less. Exhaustive and time-consuming.

Algorithm Test

Not the best method for algorithm Testing

Drawbacks

Update to automation test script is essential if you modify application frequently.

White-Box Testing

It is a method used to test a software taking into consideration its internal functioning.

Also called as structural Testing, clear box, code walk or glass box testing

Internal working is known, and the tester can test accordingly.

Testing is best suited for a lower level of testing like unit, Integration testing. Is required to perform white box testing.

~~White~~ objective of white box testing is done to check the quality of the code.

Testers and developers

Exhaustive and time consuming.

Best suited for algorithm testing

Automated test cases can become useless if the code base is rapidly changing.

Debugging Techniques

Debugging :- It is the process of finding and resolving defects or problems within a computer program that prevent correct operation of computer software or a system.

Need :- Once errors are known during a program code, it's necessary to initially establish a precise program statements liable for the errors and so to repair them.

Techniques / Approaches :-

1) Brute Force Method :- It is the most common method of debugging but is the least efficient method. In this approach, print statements are inserted throughout the program to print the intermediate values with the hope that some of the printed values will help to identify the statement in error. This approach becomes more systematic with the use of a symbolic debugger also called a source code debugger.

2) Backtracking :- This is also a fairly common approach. In this approach, starting from the statement at which an error symptom has been observed, the source code is tracked backwards until the error is discovered. Unfortunately as the no of source lines to be tracked back increases, the no of potential backward paths increases.

3) Cause elimination method :-

In this approach, once the failure is observed, the symptoms of the failure are noted. Based on the failure symptoms, the causes could have contributed to the symptom are developed and tests are conducted to eliminate each. A related technique of identification of the error from the error symptom is the software fault tree analysis.

4. Program slicing :- This technique is similar to backtracking. In the backtracking approach, one often has to examine a large no. of statements. However, the search space is reduced by defining slices. A slice of a program for a particular variable and at a particular statement is the set of source lines preceding this statement that can influence the value of that variable. Program slicing makes use of the fact that an error in the value of a variable can be caused by the statements on which it is data dependent.

Gray Box Testing

Gray Box Testing is a software testing technique to test a software product or application with partial knowledge of internal structure of the application.

→ The purpose of grey box testing is to search and identify the defects due to improper code structure or improper use of applications.

* In this process, context-specific errors that are related to the web systems are commonly identified. It increases the testing coverage by concentrating on all of the layers of any complex system.

→ Gray Box Testing is a software testing method which is a combination of both white box testing and Black Box Testing method.