**Types of Software Testing**

**INTRODUCTION:**

1. Unit Testing
2. Integration Testing
3. System Testing
4. Functional Testing
5. Acceptance Testing
6. Smoke Testing
7. Regression Testing
8. Performance Testing
9. Security Testing
10. User Acceptance Testing

Testing is the process of executing a program to find errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.

**Principles of Testing:-**

(i) All the tests should meet the customer requirements.
(ii) To make our software testing should be performed by a third party.
(iii) Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
(iv) All the tests to be conducted should be planned before implementing it
(v) It follows the Pareto rule(80/20 rule) which states that 80% of errors come from 20% of program components.
(vi) Start testing with small parts and extend it to large parts.

**Types of Testing:-**

**1. Unit Testing**

Unit testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process, where individual units of code are tested in isolation.

**The main advantages of unit testing include:**

1. It helps to identify bugs early in the development process, before they become more difficult and expensive to fix.
2. It helps to ensure that changes to the code do not introduce new bugs.
3. It makes the code more modular and easier to understand and maintain.
4. It helps to improve the overall quality and reliability of the software.

**Some popular frameworks and tools that are used for unit testing include JUnit, NUnit, and xUnit.**

It’s important to keep in mind that unit testing is only one aspect of software testing and it should be used in combination with other types of testing such as integration testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users.

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.

Example:

a) In a program we are checking if the loop, method, or

 function is working fine

b) Misunderstood or incorrect, arithmetic precedence.

c) Incorrect initialization

**2. Integration Testing**

Integration testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. Integration testing is typically done after unit testing and before functional testing, and is used to verify that the different units of the software work together as intended.

**Integration testing can be performed in different ways, such as:**

1. Top-down integration testing: It starts with the highest level modules and integrates them with lower-level modules.
2. Bottom-up integration testing: It starts with the lowest-level modules and integrates them with higher-level modules.
3. Big-Bang integration testing: It combines all the modules and integrates them all at once.
4. Incremental integration testing: It integrates the modules in small groups, testing each group as it is added.

**The main advantages of integration testing include:**

1. It helps to identify and resolve issues that may arise when different units of the software are combined.
2. It helps to ensure that the different units of the software work together as intended.
3. It helps to improve the overall reliability and stability of the software.
4. It’s important to keep in mind that Integration testing is essential for complex systems where different components are integrated together.
5. As with unit testing, integration testing is only one aspect of software testing and it should be used in combination with other types of testing such as unit testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users.

The objective is to take unit-tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

Integration testing is of four types: (i) Top-down (ii) Bottom-up (iii) Sandwich (iv) Big-Bang
Example:

(a) Black Box testing:- It is used for validation.

In this, we ignore internal working mechanisms and

focus on **what is the output?**.

(b) White box testing:- It is used for verification.

In this, we focus on internal mechanisms i.e.

**how the output is achieved?**

**3. Regression Testing**

Regression testing is a method of testing that is used to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break. It is typically done after changes have been made to the code, such as bug fixes or new features, and is used to verify that the software still works as intended.

**Regression testing can be performed in different ways, such as:**

1. **Retesting**: This involves testing the entire application or specific functionality that was affected by the changes.
2. **Re**–**execution**: This involves running a previously executed test suite to ensure that the changes did not break any existing functionality.
3. **Comparison**: This involves comparing the current version of the software with a previous version to ensure that the changes did not break any existing functionality.

**The main advantages of regression testing include:**

1. It helps to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break.
2. It helps to ensure that the software continues to work as intended after changes have been made.
3. It helps to improve the overall reliability and stability of the software.
4. It’s important to keep in mind that regression testing is an ongoing process that should be done throughout the software development
5. lifecycle to ensure that the software continues to work as intended. It should be automated as much as possible to save time and resources. Additionally, it’s important to have a well-defined regression test suite that covers

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.
Example

**4. Smoke Testing**

This test is done to make sure that the software under testing is ready or stable for further testing
It is called a smoke test as the testing of an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.
Example:

If the project has 2 modules so before going to the module

make sure that module 1 works properly

**5. Alpha Testing**

This is a type of validation testing. It is a type of *acceptance testing*which is done before the product is released to customers. It is typically done by QA people.

Example:

When software testing is performed internally within

the organization

**6. Beta Testing**

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for a limited number of users for testing in a real-time environment
Example:

When software testing is performed for the limited

number of people

**7. System Testing**

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.
In this, we have security testing, recovery testing, stress testing, and performance testing
Example:

This includes functional as well as nonfunctional

testing

**8. Stress Testing**

In this, we give unfavorable conditions to the system and check how they perform in those conditions.
Example:

(a) Test cases that require maximum memory or other

 resources are executed

(b) Test cases that may cause thrashing in a virtual

 operating system

(c) Test cases that may cause excessive disk requirement

**9. Performance Testing**

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test the speed and effectiveness of the program. It is also called load testing. In it we check, what is the performance of the system in the given load.
Example:

Checking several processor cycles.

**10. Object-Oriented Testing**

This testing is a combination of various testing techniques that help to verify and validate object-oriented software. This testing is done in the following manner:

* Testing of Requirements,
* Design and Analysis of Testing,
* Testing of Code,
* Integration testing,
* System testing,
* User Testing.

**11. Acceptance Testing**

Acceptance testing is done by the customers to check whether the delivered products perform the  desired tasks or not, as stated in requirements.

We use this OOT, for discussing test plans and for executing the projects.

This article is contributed by **Kritka**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [write.geeksforgeeks.org](http://write.geeksforgeeks.org/) or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or if you want to share more information about the topic discussed above.

**Advantages of software testing:**

1. Improved software quality and reliability
2. Early identification and fixing of defects
3. Improved customer satisfaction
4. Increased stakeholder confidence
5. Reduced maintenance costs

**Disadvantages of software testing:**

1. Time-consuming and adds to project cost
2. Can slow down development process
3. Not all defects can be found
4. Can be difficult to fully test complex systems
5. Potential for human error during testing process