**Assignment - 3**

**Name: Sammed Singalkar**

**Roll No: B-58**

**Batch: BC-4**

1. **What are the different levels of testing?**

**Ans:-**

* Unit/Component Testing
* Integration testing
* System testing
* Acceptance testing

**1. Unit/component testing**

* The most basic type of testing is unit, or component, testing.
* Unit testing aims to verify each part of the software by isolating it and then perform tests to demonstrate that each individual component is correct in terms of fulfilling requirements and the desired functionality.
* This type of testing is performed at the earliest stages of the development process, and in many cases it is executed by the developers themselves before handing the software over to the testing team.
* The advantage of detecting any errors in the software early in the day is that by doing so the team minimises software development risks, as well as time and money wasted in having to go back and undo fundamental problems in the program once it is nearly completed.

### 2. Integration testing

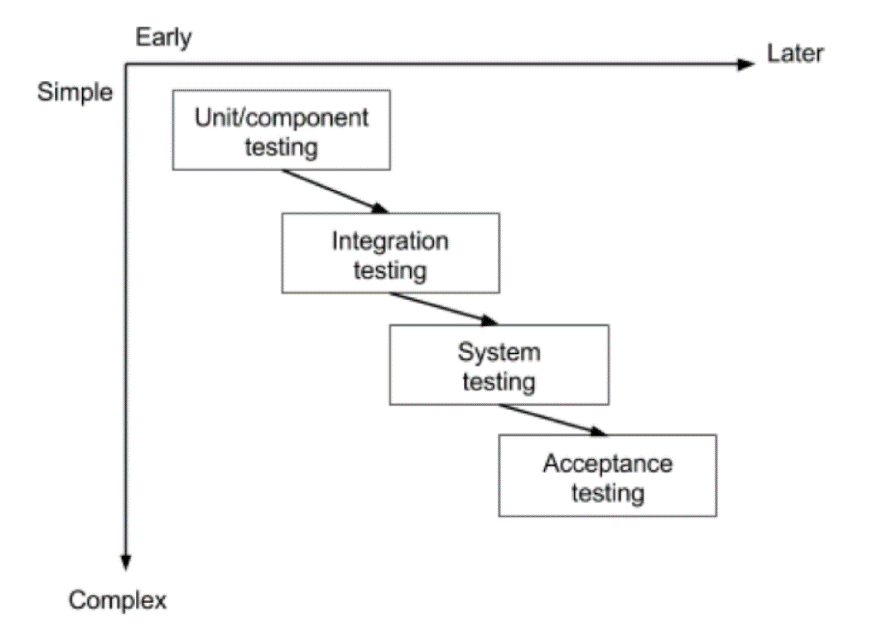
* Integration testing aims to test different parts of the system in combination in order to assess if they work correctly together. By testing the units in groups, any faults in the way they interact together can be identified.
* There are many ways to test how different components of the system function at their interface; testers can adopt either a bottom-up or a top-down integration method.
* In bottom-up [integration testing](https://en.wikipedia.org/wiki/Integration_testing), testing builds on the results of unit testing by testing higher-level combination of units (called modules) in successively more complex scenarios.
* It is recommended that testers start with this approach first, before applying the top-down approach which tests higher-level modules first and studies simpler ones later.

### 3. System testing

* The next level of testing is system testing. As the name implies, all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified.
* [System testing](https://reqtest.com/testing-blog/software-system-testing/) is a very important step as the software is almost ready to ship and it can be tested in an environment which is very close to that which the user will experience once it is deployed.
* System testing enables testers to ensure that the product meets business requirements, as well as determine that it runs smoothly within its operating environment. This type of testing is typically performed by a specialized testing team.

### 4. Acceptance testing

* Finally, [acceptance testing](https://reqtest.com/testing-blog/a-guide-to-excellent-acceptance-testing/) is the level in the software testing process where a product is given the green light or not. The aim of this type of testing is to evaluate whether the system complies with the end-user requirements and if it is ready for deployment.
* The testing team will utilise a variety of methods, such as pre-written scenarios and test cases to test the software and use the results obtained from these tools to find ways in which the system can be improved.
* The scope of acceptance testing ranges from simply finding spelling mistakes and cosmetic errors, to uncovering bugs that could cause a major error in the application.
* By performing acceptance tests, the testing team can find out how the product will perform when it is installed on the user’s system. There are also various legal and contractual reasons why acceptance testing has to be carried out.



1. **Elaborate different types of testing?**

**Ans:-**

1. Unit Testing

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.

2. Integration Testing

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.

3. Regression Testing

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.

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.

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.

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

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

8. Stress Testing

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

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.

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.