**Non Functional Testing**

Once functionality of a feature is stable, then we do non-functional Testing. **Non Functional Testing** mainly focuses on customer expectations not requirements.

In E Commerce applications, the features such as login, order, cart these are customer requirements. While doing Functional testing, we are mainly focussing on requirements and checking whether functionality is working correctly or not.

Now lets say Functionality of application is working fine, but what if the page is taking too long to respond, performance is low, it is not secured. Now these are the customer expectations we are speaking about. We are checking if the application is good, it is secured, it is working on different platforms, it is safe from unauthorized users.

Example: If more users use the application at the same time, we should ensure the application does not go slow.

Non-Functional Testing done by a separate team.

· **Performance Testing**

Performance Testing deals with checking the speed of an application. It is done for Web applications, but not applicable for Desktop applications.

**There are 3 types of Performance Testing:**

1) **Load Testing** – Increasing the load(multiple users) of the application slowly and checking the speed of the application. We use tools(jmeter, loadrunner) which will create virtual users to perform these tasks.

2) **Stress Testing** – Here we suddenly increase or decrease the load of the application and check the speed, stability of the application.

3) **Volume Testing** – Here we check how much data or volume an application can handle, we can check through volume testing.

**Importance of Performance Testing:**

**● Performance** Testing contributes to user satisfaction. Users expect applications to be fast and reliable. Through this testing we can achieve customer satisfaction. ● Performance testing verifies the stability of the application under different levels of load. Applications should remain stable and reliable even when subjected to heavy user traffic or data processing loads.

● Performance testing validates the speed and efficiency of critical functionalities. Applications with slow or inefficient performance can deter users and impact the

competitiveness of the software.

● Performance testing helps identify potential issues that could lead to system downtime. Downtime can have severe consequences, including loss of revenue, damage to reputation, and inconvenience to users.

● Performance testing reduces the risk of performance-related failures in production.

Testing the functionality of an application in different hardware and software configurations or platforms or environments is called Compatibility Testing.

**Q) Why is compatibility Testing important?**

● If TE tests the software in one platform and release to market, customers can use the software in different platforms and software might not work as expected in other platforms, because of this customer market value goes down, to avoid this Compatibility Testing is a must.

● To ensure that all features are consistently working in all platforms,

compatibility testing is necessary.

**Compatibility Testing**

Testing the functionality of an application in different hardware and software configurations or platforms or environments is called Compatibility Testing.

**Q) Why is compatibility Testing important?**

● If TE tests the software in one platform and release to market, customers can use the software in different platforms and software might not work as expected in other platforms, because of this customer market value goes down, to avoid this Compatibility Testing is a must.

● To ensure that all features are consistently working in all platforms,

compatibility testing is necessary.

**Q) How to do Compatibility Testing?**

A) First we need to identify the base platform, I,e,, based on market research we need to analyze which browser, OS, MAC, Android versions most users are using. Which platform is widely used by our customers, that is our base platform. First we need to do all types of testing in our base platforms, the perform all testing in other platforms.

Note: Developers while writing the code can either write it in platform specific code(which is specific to particular platform) or can write it as Generic code(Test in all platforms). Type of defects we see usually while doing compatibility Testing:

1) Scattered content defect

2) Alignment issues

3) Object overlapping

4) Look and feel of the application

**Important Differences between Functional and Non-Functional Testing:**

**Functional Testing**

1) It is done before Non-Functional Testing

2) It validates/checks functionality of the software

3) Functionality describes what software does

4) Works on customer requirements

**Non-Functional Testing**

1) It is done after Functional Testing

2) It validates/checks non- functionality of the software

3) Functionality describes how software works

4) Works on customer expectations

**Whitebox testing**

● Focus: White box testing examines the internal structure, logic, and code of the software.

● Knowledge: Testers require knowledge of the software's source code and internal architecture.

● Objective: The primary goal is to ensure that the code is implemented correctly, addressing code quality, logic errors, and code coverage.

● Test Design Techniques: White box testing techniques include statement coverage, branch coverage, path coverage, and

code walkthroughs.

● Test Cases: Test cases are designed based on the

software's internal structure, code paths, and logic.

White-box testing, also known as structural testing or glass-box testing, involves testing the internal logic and structure of a software application. In white-box testing, the tester has access to the source code and uses this knowledge to design test cases that ensure thorough coverage of the code. Several techniques are employed for white-box test case design:

**1. Statement Coverage**

**2. Branch Coverage**

**3. Path Coverage**

**4. Condition Coverage**

**5. Loop Testing**

**6. Data Flow Testing**

White-box testing techniques are complementary to black-box testing techniques, providing a thorough examination of the internal workings of the software. These techniques help ensure that the code is not only functionally correct but also robust and resilient to various inputs and conditions.

**GREYBOX TESTING DESIGN TECHNIQUES**

Focus: Grey box testing is a combination of both black box and white box testing. Testers have partial knowledge of the internal code and structure.

Knowledge: Testers may have access to limited information about the software's internals, such as high-level architecture or data flow diagrams.

Objective: The goal is to assess both the external behaviour and internal logic of the software, making it suitable for integration testing, security testing, and vulnerability assessments. Test Design Techniques: Grey box testing combines techniques from both black box and white box testing, allowing for a more holistic approach to test design.

Test Cases: Test cases may be based on a combination of requirements and knowledge of the software's architecture and code.