**Functional testing**

Testing the features and functionality of the software (input, output, and user interface), aim of the testing is that the software meets all the requirements specified in the requirements document.

Functional testing checks the application's processes against a set of requirements or specifications.

Mainline function

Functional testing

Basic usability

Basic accessibility

Error conditions

**Example**

* Testing the shopping website, testing would involve whether the website can perform task such as adding items to the card, apply discount & processing payment correctly, also check the website is user friendly & link button and menu all are work as intended

1- Its test the functionality of the software

2-It’s based on customer requirement

3-Ex- Login page must show text boxes to enter the user name and password

* The app wants to show users nearby food restaurant based on their GPS data. Verifying this function requires a module test, as it requires multiple units to work together to get the expected result.

**Types of functional testing**

Unit testing (which can also be used for non-functional testing)

Integration testing

User acceptance testing

Closed-box testing

**Advantages of Functional Testing**

It makes the program complies with the functional specifications listed in the design or requirements documentation

Reduces the cost and difficulty of repairing software faults

Problems later on by finding them early in the development cycle.

It assures users that the software will work as intended and meet their expectations.

Ensuring that the user interface, input/output, and error handling correctly contributes to enhancing the user experience.

Ensures that the software is of a high standard and satisfies the requirements of both the users and the business.

**Nonfunctional testing**

When the software works as per the user expectations, smoothly & efficiently under any condition. Non-functional testing assesses application properties that aren't critical to functionality but contribute to the end-user experience, like performance and reliability under load.

Performance testing

Security testing

Interoperability testing

Nonfunctional testing

Scalability testing

Reliability testing

Usability testing

**Example:**

* An application is developed and it’s complete it’s functional testing, but nonfunctional test not tested on the same. When the application goes live, if its load increase, it becomes too slot and take lot of time to open.

1- Its test the performance of the functionality of software

2-It’s based on customer expectation

3-Ex- us enter in to the login page it getting loaded in 5 seconds.

* An ecommerce app must be optimized to handle heavy user traffic for the upcoming sale days. Load tests, recovery tests, and volume tests must be run to ensure that it does not buckle and bend under a wildly increased inflow of users.

**Types of non-functional tests**

Performance testing

Load testing

Usability testing

Security testing

**Advantages of Non-Functional Testing**

Ensuring the program satisfies the non-functional requirements, including performance, security, usability, reliability, and compatibility.

Detects performance stumbling blocks, security holes, usability problems, and other non-functional flaws before they affect end users.

Ensuring the software is responsive, scalable, and user-friendly enhances the user experience.

Ensures that the program complies with rules and standards set by the industry.

Increases trust in the software's performance in real-world circumstances and under different loads and conditions.