Functional Testing and Non-Functional Testing

Functional Testing and Non-Functional Testing are two different categories of software testing, each serving a distinct purpose in evaluating the quality and performance of a software system.

1. Purpose:

- **Functional Testing:** This type of testing assesses the functionality of a software application and verifies whether it performs its intended operations correctly. It focuses on "what" the software should do.
- **Non-Functional Testing:** Non-functional testing evaluates aspects of the software that are not directly related to its functionality but are essential for overall user experience and performance. It focuses on "how" the software performs.

2. What They Test:

- **Functional Testing:** It tests specific functions or features of the software. This includes testing user interfaces, data processing, input validation, and business logic. Examples of functional tests include login functionality, form submissions, and data retrieval.
- **Non-Functional Testing:** Non-functional testing assesses aspects like performance, scalability, security, reliability, usability, and compatibility. Examples of non-functional tests include load testing, security testing, usability testing, and performance testing.

3. Test Focus:

- **Functional Testing:** It is concerned with validating that the software meets the specified requirements and that it performs its functions correctly. It often involves positive and negative testing of individual features.
- **Non-Functional Testing:** Non-functional testing focuses on evaluating attributes that affect the overall user experience, such as response times, security measures, and user interface usability.

4. Examples of Functional Testing:

1. **Unit Testing:** Tests individual components or units of code in isolation. For example, testing a function that calculates the total price of items in a shopping cart.

2. **Integration Testing:** Tests how different units or modules work together when integrated. For instance, testing how a user authentication module interacts with a database

3. **System Testing:** Evaluates the entire system to ensure it meets the specified requirements. It involves testing end-to-end scenarios, like the entire e-commerce checkout process.

Functional Testing and Non-Functional Testing

4. **User Acceptance Testing (UAT):** Allows end-users to validate that the software meets their needs and expectations by testing real-world scenarios.

5. Examples of Non-Functional Testing:

1. **Performance Testing:** Ensures that the system performs efficiently and meets response time requirements. For example, load testing to determine how the system handles a specific number of concurrent users.

2. **Security Testing:** Identifies vulnerabilities and assesses the system's security measures, such as penetration testing and vulnerability scanning.

3. **Usability Testing:** Evaluates the user-friendliness and overall user experience of the software, including interface design, navigation, and accessibility.

4. **Reliability Testing:** Assesses the software's stability and availability under various conditions, such as stress testing to see how it handles extreme loads.

5. **Compatibility Testing:** Ensures that the software functions correctly across different devices, browsers, and operating systems.