Functional and Non functional testing

What is functional testing?

Functional testing is a type of software testing that verifies whether the software application behaves according to its specified functional requirements. It focuses on what the system does rather than how it performs internally. The purpose is to ensure that all features and functionalities work as expected, validating the correctness of outputs based on given inputs. Functional testing is primarily requirement-driven and often performed using black-box testing techniques, where the internal code structure is not considered.

Purpose

- To ensure that the software performs its intended functions correctly.
- To validate the output against the expected result for given inputs.
- To catch defects or discrepancies in the functionality.

Key Characteristics

- **Requirement-based:** Tests are derived from functional requirements or specifications.
- **Black-box approach:** Testers don't need to know the internal code or implementation.
- **Focus on user actions:** Checks user workflows, inputs, outputs, and system responses.

Common Functional Testing Activities

- **Unit Testing:** Tests individual functions or modules (sometimes considered more developer-level).
- **Integration Testing:** Ensures that different modules or components work together correctly.
- **System Testing:** Validates the complete and integrated application.
- **User Acceptance Testing (UAT):** Confirms that the system meets business requirements from the user's perspective.

What is non functional testing?

Non-functional testing is a type of software testing that evaluates the quality attributes of a system, such as performance, reliability, usability, scalability, and security. Unlike functional testing, it focuses on how well the system performs under different conditions rather than on specific features. The purpose is to ensure the software meets user expectations for speed, stability, security, and user experience. Non-functional testing often involves stress, load, and usability assessments to validate the system's behavior in real-world scenarios.

Purpose

- **Assess Performance:** Ensure the system is fast, stable, and responsive under load.
- **Check Security:** Validate that the system protects data and prevents unauthorized access.
- **Evaluate Usability:** Confirm the software is user-friendly and easy to navigate.
- **Test Reliability & Scalability:** Ensure the system can handle growth and continues working under stress.

Key Characteristics

Non-functional testing evaluates software quality attributes like performance, security, usability, and reliability.

It focuses on how well the system behaves under various conditions rather than on specific functionality.

It ensures the system can handle load, scale effectively, remain compatible across devices, and provide a good user experience.

Common Non-Functional Testing Activities

1. **Performance Testing** – Measures how the system performs under normal and peak loads to ensure it responds quickly and operates efficiently.

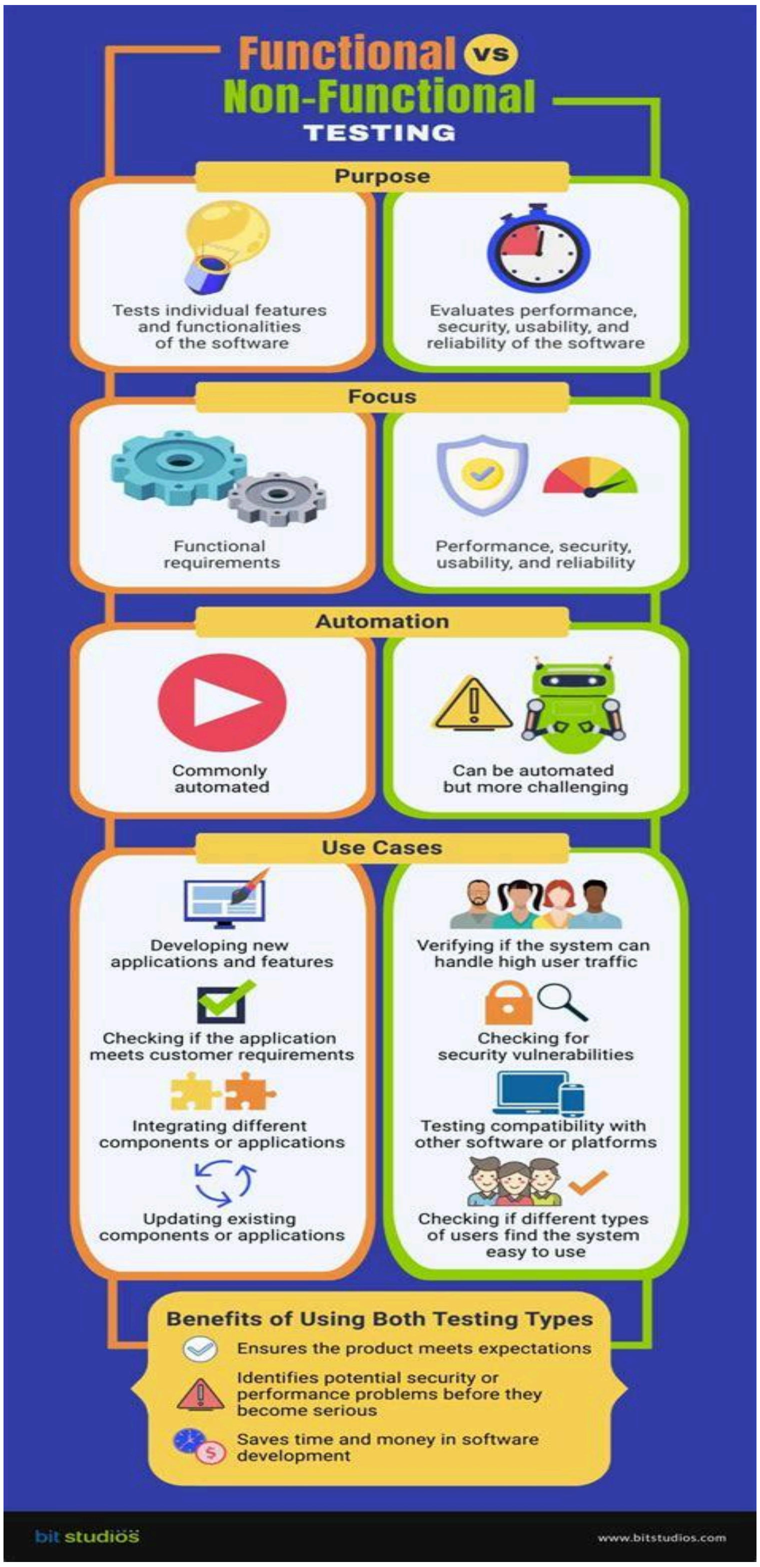
2. **Load Testing** – Checks the application's behavior when multiple users access it simultaneously to ensure stability.

3. **Stress Testing** – Tests the system under extreme conditions or beyond its limits to see how it recovers from failure.

4. **Usability Testing** – Evaluates how easy and intuitive the software is for end users, including navigation, layout, and accessibility.

5. **Security Testing** – Ensures the system is protected against unauthorized access, data breaches, and vulnerabilities.

Functional Testing	**Non-Functional Testing**
Testing that verifies whether the software performs its intended functions correctly according to requirements.	Testing that evaluates how well the system performs, focusing on quality attributes like performance, security, and usability.
Checks what the software does (specific features and functions).	Checks how well the software performs under various conditions.
Ensure core functionalities work as expected.	Ensure the system is efficient, reliable, secure, and user-friendly.
Black-box testing: focuses on inputs, outputs, and workflows without looking at internal code.	Behavior-oriented: tests performance, usability, scalability, and reliability.
Login, registration, shopping cart, form submission.	Load time, stress handling, security, compatibility, usability.



Functional vs Non-Functional TESTING

	Functional	Non-Functional
Purpose	Tests individual features and functionalities of the software	Evaluates performance, security, usability, and reliability of the software
Focus	Functional requirements	Performance, security, usability, and reliability
Automation	Commonly automated	Can be automated but more challenging
Use Cases	Developing new applications and features	Verifying if the system can handle high user traffic
	Checking if the application meets customer requirements	Checking for security vulnerabilities
	Integrating different components or applications	Testing compatibility with other software or platforms
	Updating existing components or applications	Checking if different types of users find the system easy to use

Benefits of Using Both Testing Types

- Ensures the product meets expectations
- Identifies potential security or performance problems before they become serious
- Saves time and money in software development

bit studios

www.bitstudios.com