



TYPES OF TESTING

Mansi Thakur

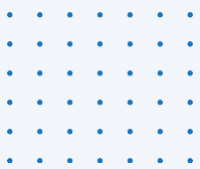


Table of Contents

1. Functional Testing	1
1.1 Smoke Testing.....	1
1.2 Sanity Testing.....	1
1.3 User Acceptance Testing (UAT).....	1
1.4 Regression Testing	1
2. Non-Functional Testing	1
2.1 Performance Testing.....	1
2.1.1 Types of Performance Testing	2
2.2 Security Testing	2
2.2.1 Types of Security Testing	2
2.3 Usability Testing	2
2.4 Compatibility Testing.....	2
3. Other Types of Testing.....	2
3.1 Alpha Testing	2
3.2 Beta Testing	3
3.3 Mutation Testing	3

Testing is a key activity in the software development lifecycle. It ensures that the software meets its requirements, behaves as expected, and is free from defects. The testing process is classified into various types, each serving a specific purpose. The different types of testing can be broadly categorized based on various factors, such as the phase of testing, the aspect being tested, and the method of testing.

1. Functional Testing

Functional testing focuses on verifying whether the software performs according to the specified requirements. These tests are designed to validate the system's behavior and functionality.

1.1 Smoke Testing

- **Purpose:** Ensures that the basic functionalities of the system are working after a new build or release.
- **Scope:** Focuses on critical functionalities that the application depends on.
- **When to Use:** When a new build is deployed or during initial testing phases.
- **How to Perform:** Execute a minimal set of tests to check if the application is stable enough for further testing.

1.2 Sanity Testing

- **Purpose:** Verifies that specific functionalities or features work as expected after a fix or change is made.
- **Scope:** Narrower than smoke testing, focusing on particular features.
- **When to Use:** After bug fixes or minor changes to ensure the fix has not impacted other parts of the system.
- **How to Perform:** Test only the modified or newly added features and ensure they work correctly.

1.3 User Acceptance Testing (UAT)

- **Purpose:** Ensures that the software meets business requirements and is ready for production.
- **Scope:** Focuses on real-world use cases and user perspectives.
- **When to Use:** After system testing and before release to production.
- **How to Perform:** Business users test the system against business scenarios and provide feedback.

1.4 Regression Testing

- **Purpose:** Ensures that new changes have not broken existing functionalities.
- **Scope:** Whole system or impacted modules.
- **When to Use:** After code changes, bug fixes, or enhancements.
- **How to Perform:** Re-run previous test cases and focus on areas that may be impacted by recent changes.

2. Non-Functional Testing

Non-functional testing focuses on aspects of the software that are not related to specific functionalities, such as performance, security, usability, and compatibility.

2.1 Performance Testing

- **Purpose:** Assesses the speed, scalability, and stability of the system under varying load conditions.
- **Scope:** Includes load testing, stress testing, and scalability testing.
- **When to Use:** To ensure the system can handle expected traffic and user activity.

- **How to Perform:** Simulate user load to test how the system performs under normal and extreme conditions.

2.1.1 Types of Performance Testing

- **Load Testing:** Verifies the system can handle the expected load (number of users, requests).
- **Stress Testing:** Evaluates system behavior under extreme load conditions, such as very high traffic or resource shortages.
- **Scalability Testing:** Tests the system's ability to scale (e.g., adding more servers) under increased load.

2.2 Security Testing

- **Purpose:** Identifies vulnerabilities in the system and ensures that data is protected from threats.
- **Scope:** Focuses on encryption, authentication, authorization, and data protection.
- **When to Use:** Before the system is deployed or after major changes in security requirements.
- **How to Perform:** Conduct penetration testing, vulnerability scanning, and check for security standards compliance.

2.2.1 Types of Security Testing

- **Penetration Testing:** Simulates an attack on the system to identify vulnerabilities.
- **Vulnerability Scanning:** Scans the application for known security flaws.
- **Compliance Testing:** Ensures the application complies with industry standards (e.g., HIPAA, GDPR).

2.3 Usability Testing

- **Purpose:** Evaluates the user interface (UI) and user experience (UX) to ensure the system is easy to use and intuitive.
- **Scope:** Focuses on design, navigation, accessibility, and overall user satisfaction.
- **When to Use:** After system functionality is verified, typically before release.
- **How to Perform:** Users interact with the system while testers observe, gather feedback, and identify potential usability issues.

2.4 Compatibility Testing

- **Purpose:** Ensures that the software works across different environments, browsers, devices, and operating systems.
- **Scope:** Focuses on cross-platform, cross-browser, and mobile compatibility.
- **When to Use:** When you want to ensure that your application works seamlessly across different environments.
- **How to Perform:** Test on various browsers (Chrome, Firefox, Safari), devices (mobile, tablet, desktop), and operating systems (Windows, macOS, Linux).

3. Other Types of Testing

Some tests focus on specific requirements or business needs.

3.1 Alpha Testing

- **Purpose:** Performed by developers or an internal QA team to identify issues before releasing the product to external users.
- **Scope:** In-house testing, typically done at the development site.
- **When to Use:** Before beta testing, usually before releasing to a limited external audience.
- **How to Perform:** Testers simulate real-world use cases and check for issues.

3.2 Beta Testing

- **Purpose:** Performed by a selected group of external users to get feedback and identify potential issues before the final release.
- **Scope:** Performed in a real-world environment by a limited audience.
- **When to Use:** After alpha testing, just before the final release.
- **How to Perform:** Provide the software to users, gather feedback, and fix reported issues.

3.3 Mutation Testing

- **Purpose:** Evaluates the effectiveness of existing tests by introducing small changes (mutations) to the code.
- **Scope:** Test suite effectiveness.
- **When to Use:** To assess the robustness and coverage of test cases.
- **How to Perform:** Modify the code slightly (e.g., change conditions or logic) and run tests to see if they catch the errors.