**What is software Testing**

Software testing is a crucial phase in the software development life cycle (SDLC). It ensures the software functions correctly, meets user requirements, and is free from defects. Below are the most common types of testing with examples.

**Types of Testing with Examples**

**1. Smoke Testing**

Smoke testing Ensures the core functionalities work after a new build or deployment.

* Example: After deploying a new version of an e-commerce website, testers verify that the login page, homepage, and product browsing are functional.

2. Sanity Testing

Sanity Testing Focuses on verifying specific bug fixes or minor changes without full regression testing.

* Example: After fixing a payment gateway issue, testers only verify the payment process to ensure it's working correctly.

3. Regression Testing

Regression Testing Ensures new code changes haven’t affected existing functionalities.

* Example: After adding a 'Wishlist' feature, testers ensure login, cart, and checkout still work properly.

4. Functional Testing

Functional Testing verifies the application's functionality according to requirements.

* Example: Testing the "Forgot Password" feature to ensure users receive a password reset email.

5. Integration Testing

. Integration Testing ensures different modules work together seamlessly.

* Example: Testing if adding an item to the cart updates the total price correctly.

6. System Testing

System Testing validates the entire system's functionality in an environment similar to production.

* Example: Testing an online food delivery app's end-to-end process from order placement to payment confirmation.

7. User Acceptance Testing (UAT)

UAT Ensures the software meets business requirements before release.

* Example: A client verifies the final e-commerce platform's checkout process, ensuring it's user-friendly and error-free.

8. Performance Testing

Performance Testing evaluates the system’s speed, responsiveness, and stability under various loads.

* Example: Testing a banking app to ensure it handles 1,000 simultaneous transactions efficiently.

9. Security Testing

Security Testing identifies vulnerabilities to protect the system from threats.

* Example: Testing a login page by attempting SQL injection to ensure data security.

10. Exploratory Testing

Exploratory Testing Involves testers exploring the application freely to uncover unexpected issues.

* Example: Randomly clicking through different features in a new app to identify hidden bugs.

11. Compatibility Testing

Compatibility Testing ensures the application works correctly across different devices, browsers, and operating systems.

* Example: Testing a website on Chrome, Firefox, and Safari to ensure consistent layout and functionality.

12. Localization Testing

Localization Testing ensures the application adapts to different languages, cultures, and regions.

* Example: Checking that a shopping app displays product prices in INR for Indian users and USD for American users.

13. Usability Testing

Usability Testing Ensures the application is user-friendly and intuitive.

* Example: Testing a food delivery app’s navigation to ensure users can easily place an order.

14. Recovery Testing

Recovery Testing verifies the system's ability to recover from failures.

* Example: Testing if a banking app resumes session data after an unexpected crash.

15. Installation Testing

Installation Testing ensures the software installs, updates, and uninstalls properly.

* Example: Verifying that a photo editing software installs without errors and uninstalls completely.

16. Compliance Testing

Compliance Testing ensures the software adheres to industry standards and regulations.

* Example: Testing a healthcare application to ensure it complies with HIPAA data protection standards.