Software Testing Life Cycle (STLC)

STLC is the process followed during testing to ensure software quality. It is a systematic approach that consists of several phases, each with specific tasks and deliverables.

Phases of STLC

1. **Requirement Analysis**
    - **Objective**: Understand what to test.
    - **Example**: Reviewing a requirement document to verify if the login page should allow users to reset forgotten passwords.
2. **Test Planning**
    - **Objective**: Plan the testing strategy, scope, schedule, and resources.
    - **Example**: Deciding to perform functional and performance testing for an e-commerce website.
3. **Test Case Design**
    - **Objective**: Write detailed test cases based on requirements.
    - **Example**: Writing a test case to verify if the "Add to Cart" button adds an item to the shopping cart.
4. **Test Environment Setup**
    - **Objective**: Prepare the test environment to execute test cases.
    - **Example**: Setting up a database and deploying the application on a staging server.
5. **Test Execution**
    - **Objective**: Run test cases and report defects.
    - **Example**: Testing the login feature by entering valid and invalid credentials.
6. **Defect Reporting and Retesting**
    - **Objective**: Log defects, fix them, and retest the functionality.
    - **Example**: Reporting that the "Forgot Password" link does not send a reset email.
7. **Test Closure**
    - **Objective**: Evaluate test completion, document lessons learned, and archive test artifacts.
    - **Example**: Preparing a test summary report indicating all critical bugs have been resolved.

Types of Testing with Simple Examples

1. Functional Testing

- **Purpose**: Verify that software functions as expected.
- **Example**: Checking if entering valid credentials successfully logs the user in.

2. Non-Functional Testing

- **Purpose**: Evaluate performance, usability, and reliability.
- **Example**: Measuring the time it takes to load a webpage under heavy traffic.

3. Unit Testing

- **Purpose**: Test individual components or functions.
- **Example**: Verifying that a function calculating total cart value returns the correct sum.

4. Integration Testing

- **Purpose**: Test the interaction between integrated modules.
- **Example**: Ensuring that the checkout page fetches correct product details from the database.

5. System Testing

- **Purpose**: Validate the entire system against requirements.
- **Example**: Testing the end-to-end flow of an e-commerce site, from browsing products to payment.

6. Acceptance Testing

- **Purpose**: Verify if the application meets business requirements and is ready for release.
- **Example**: A client testing the application to ensure it satisfies all user stories.

7. Regression Testing

- **Purpose**: Check if new changes have affected existing features.
- **Example**: Testing the login feature after updating the UI to ensure functionality is unaffected.

8. Smoke Testing

- **Purpose**: Verify basic functionality in a new build.
- **Example**: Confirming that the application launches and the homepage loads.

9. Sanity Testing

- **Purpose**: Verify specific functionality after a minor change.
- **Example**: Testing if a recently fixed "Search" feature works without testing the entire app.

10. Performance Testing

- **Purpose**: Measure speed, scalability, and stability.

- **Example**: Checking how many users can simultaneously log in without crashing the system.

11. Load Testing

- **Purpose**: Determine how the application performs under expected load.
- **Example**: Testing if a website can handle 1,000 users browsing products at the same time.

12. Stress Testing

- **Purpose**: Test application limits beyond normal conditions.
- **Example**: Simulating 10,000 users accessing the system to observe failure points.

13. Usability Testing

- **Purpose**: Check how user-friendly the application is.
- **Example**: Observing if users can easily navigate an online shopping platform.

14. Security Testing

- **Purpose**: Ensure application safety from vulnerabilities.
- **Example**: Verifying if unauthorized users cannot access admin panels.

15. Compatibility Testing

- **Purpose**: Check if the application works across devices and browsers.
- **Example**: Testing a website on Chrome, Firefox, and Safari on different operating systems.

16. Exploratory Testing

- **Purpose**: Test without predefined test cases, using creativity and domain knowledge.
- **Example**: Randomly clicking links on a website to find hidden issues.

17. Alpha Testing

- **Purpose**: Conducted by internal teams to catch bugs before releasing to customers.
- **Example**: Developers and testers using the application to identify issues.

18. Beta Testing

- **Purpose**: Involves real users testing the application in a real environment.
- **Example**: Launching an app to a small group of users for feedback.