Introduction to Software Testing

- Software testing ensures that software functions correctly and meets specified requirements.

Types of Software Testing

- Unit testing, integration testing, system testing, and acceptance testing are common types.

Unit Testing

- Tests individual components of the software to ensure they work in isolation.

Integration Testing

- Ensures that different modules or services work together as expected.

System Testing

- Verifies that the complete software system meets the specified requirements.

Acceptance Testing

- Ensures the software meets the business needs and is ready for release.

Static Testing

- Involves reviewing documents and code without executing the software.

Dynamic Testing

- Involves executing code and checking the output for expected results.

White Box Testing

- The tester has knowledge of the internal code and tests the logic directly.

Black Box Testing

- Tests functionality without knowing the internal workings of the software.

Automated Testing

- Uses software tools to run tests automatically, improving efficiency.

Manual Testing

- Human testers manually execute tests and review the results.

Test Cases

- Test cases define the inputs, execution conditions, and expected results for a test.

Test Plans

- A test plan outlines the testing strategy, scope, and resources needed for testing.

Test Scripts

- A set of instructions that are executed during testing, especially in automated tests.

Test Data

- Data used during testing to check the system's behavior under various conditions.

Regression Testing

- Ensures that new code changes do not break existing functionality.

Performance Testing

- Tests how the system performs under high load or stress conditions.

Load Testing

- Measures the system's ability to handle expected user loads.

Stress Testing

- Puts the system under extreme conditions to test its limits.

Usability Testing

- Ensures the software is user-friendly and meets the needs of its users.

Security Testing

- Identifies vulnerabilities and ensures that the system is secure against attacks.

Alpha Testing

- Conducted by internal developers or testers before the software is released.

Beta Testing

- Performed by end users in a real-world environment to catch issues before the final release.

Test Metrics

- Metrics like pass/fail rates help measure the effectiveness of testing.

Test Automation Tools

- Popular tools include Selenium, JUnit, and Jenkins for automating the testing process.

Bug Tracking

- Tracking and managing bugs is critical to fixing defects in a timely manner.

Review Techniques

- Reviews help identify errors early and improve software quality.

Continuous Integration & Testing

- Automating the integration and testing of code helps detect issues early.

Conclusion: Importance of Testing

- Software testing is vital to delivering a high-quality product.