Topics Covered:

- Software Testing Life cycle
- Types of Testing
- Write test cases

Software Testing life cycle

1. **Requirement Analysis** – Understanding project requirements and defining testable conditions.

    - Identify testable and non-testable requirements.
    - Engage with stakeholders to clarify doubts.
    - Define acceptance criteria.

2. **Test Planning** – Defining the scope, objectives, and strategy for testing.

    - Prepare a test plan document.
    - Estimate resources, schedule, and budget.
    - Identify risks and mitigation strategies.

3. **Test Case Development** – Designing and writing test cases.

    - Develop test scenarios and test cases.
    - Prepare test data based on requirements.
    - Conduct peer reviews for validation.

4. **Test Environment Setup** – Configuring the test environment.

    - Identify required hardware, software, and network configurations.
    - Set up test environments for different testing types.
    - Validate the test environment before execution.

5. **Test Execution** – Running test cases and reporting defects.

    - Execute test cases manually or using automation tools.
    - Log defects in Jira for failed test cases.
    - Retest defects after fixes and perform regression testing.

6. **Test Closure** – Evaluating test results and preparing closure reports.

    - Ensure test coverage is adequate.
    - Conduct retrospective meetings.
    - Archive test cases, test reports, and defect logs for future reference.

Types of Testing

- **Unit Testing**: Testing individual components or functions.
- **Integration Testing**: Checking interactions between components.
- **System Testing**: Evaluating the entire system against requirements.
- **Acceptance Testing**: Verifying the product meets business needs.
- **Regression Testing**: Ensuring recent changes haven't introduced new issues.
- **Performance Testing**: Measuring responsiveness and stability under load.
- **Security Testing**: Identifying vulnerabilities in the system.
- **Usability Testing**: Ensuring the application is user-friendly.

Writing Test Cases for a Sample Application

1. Select a sample application (e.g., login page, or search functionality).
2. Write detailed test cases, including:
    - **Test Case ID**
    - **Test Scenario**
    - **Test Steps**
    - **Test Data** (if applicable)
    - **Expected Result**
    - **Actual Result** (during execution)
    - **Pass/Fail Status**
3. Ensure test cases cover positive, negative, boundary, and edge cases.
4. Categorize test cases based on priority and severity.

Examples

---

1. **Write test cases for login page (5 test cases only)**

Test Case 1: Verify Successful Login with Valid Credentials

- **Test Case ID**: TC_LP_001
- **Test Scenario**: Verify that a user can log in successfully with valid credentials.
- **Test Steps**:
    1. Open the login page.
    2. Enter a valid username.
    3. Enter a valid password.
    4. Click the "Login" button.
- **Test Data**: Username: `valid_user`, Password: `valid_pass123`
- **Expected Result**: The user is redirected to the homepage/dashboard.
- **Actual Result**: (To be filled after execution)
- **Status**: Pass/Fail

Assignment (deadline 25 Feb 2025)

- **Write test cases for search functionality (10 test cases)**
- **Make a report contains the difference between Test Cases, Edge Cases, and Boundary Cases.**

Submission on LMS.