# **Tester Roles and Responsibilities in Software Development**

## **Overview**

A Tester (or QA Engineer) ensures the quality of the software by verifying that the POST /api/users endpoint functions as per requirements, is free of defects, and meets business standards. In the Spring Boot application, testers focus on validating user creation, including input validation and error handling.

## **Detailed Responsibilities**

### **1. Requirement Review and Clarification**

* **Activities**:
  + Collaborate with BA and developers to understand POST /api/users requirements.
  + Example: Confirm that email must be unique, age is 18-120, phone follows +84/0 + 9-10 digits.
  + Identify ambiguous requirements for clarification.
* **Deliverables**:
  + Notes on unclear requirements.
* **In Project**: Understand validation annotations in User.java (@NotBlank, @Email, @Min, @Pattern).

### **2. Test Planning**

* **Activities**:
  + Create a Test Plan outlining testing strategy, scope, resources, and schedule.
  + Example: Plan to test request validation, response codes, and database storage.
* **Deliverables**:
  + Test Plan document.
* **In Project**: Define testing scope for POST /api/users, including all validation scenarios.

### **3. Test Case Design and Creation**

* **Activities**:
  + Design test cases covering happy, negative, and boundary scenarios.
  + Example:
    - Happy: Valid user data.
    - Negative: Invalid email, age < 18.
    - Boundary: Age = 18 or 120.
* **Deliverables**:
  + Test case document (see TestCases\_POST\_api\_users.xlsx).
* **In Project**: Create test cases for all User.java validations.

### **4. Test Environment and Data Preparation**

* **Activities**:
  + Set up the test environment (Spring Boot app on localhost:8080, H2 database).
  + Prepare test data (valid and invalid inputs).
* **Deliverables**:
  + Configured test environment and test data.
* **In Project**: Ensure H2 database is empty before tests to avoid unique email conflicts.

### **5. Test Execution**

* **Activities**:
  + Execute test cases manually (via Postman) or automatically (via REST-assured).
  + Example: Send POST /api/users with invalid email and verify 400 response.
* **Deliverables**:
  + Test execution results (pass/fail).
* **In Project**: Verify endpoint returns correct status and data.

### **6. Defect Reporting and Management**

* **Activities**:
  + Log defects in JIRA with details (description, steps to reproduce, severity).
  + Example: Report if endpoint accepts invalid email without error.
* **Deliverables**:
  + Defect tickets.
* **In Project**: Report issues with validation logic (e.g., @Email not enforced).

### **7. Retesting and Regression Testing**

* **Activities**:
  + Retest fixed defects to confirm resolution.
  + Run regression tests to ensure no new issues.
* **Deliverables**:
  + Retest and regression test results.
* **In Project**: Re-run test cases after fixes to ensure POST /api/users stability.

### **8. Collaboration with Development Team**

* **Activities**:
  + Work with developers to clarify defects and requirements.
  + Example: Discuss why endpoint returns 500 instead of 400.
* **Deliverables**:
  + Resolved defects and improved communication.
* **In Project**: Ensure @Valid in UserController.java works as expected.

## **Required Skills**

* **Analytical Skills**: Design comprehensive test cases based on requirements.
* **Technical Skills**: Use Postman, REST-assured, JIRA, SQL (H2 database).
* **Communication Skills**: Report defects clearly and collaborate with team.
* **Tools**: Postman, JIRA, Excel, REST-assured, H2 Console.
* **Methodologies**: Agile, manual testing, automation testing.

## **References**

* The Knowledge Academy, Alphabin, Testomat.io, and other industry sources