Behaviour Driven Development (BDD) with Cucumber and Gherkin

---

1. BDD – Behaviour Driven Development

Introduction

- **BDD** is a **software development approach** that encourages **collaboration between developers, testers, and business stakeholders**.
- It focuses on **defining system behavior** using simple **natural language**, making requirements easier to understand.

Key Features:

- Uses **examples (scenarios)** to describe system behavior.
- Bridges the gap between **technical and non-technical team members**.
- Relies on **Gherkin language** to describe scenarios.
- Ensures **living documentation** that evolves with the project.

---

2. Agile and BDD Framework

- **Agile** delivers software iteratively and incrementally.
- **BDD fits perfectly with Agile** because:
    - Requirements are clarified early.
    - Test cases (scenarios) double as documentation.
    - Promotes **continuous collaboration**.
    - Supports **test automation** through tools like Cucumber.

---

3. Three Amigos Session

- A **BDD practice** where 3 roles come together:
    - **Business Analyst / Product Owner (PO)** – defines business value.
    - **Developer** – ensures feasibility.
    - **Tester (QA)** – validates acceptance criteria.
- **Outcome:** Well-defined scenarios that everyone agrees on.

Example:

- **User Story:** As a customer, I want to log in so that I can access my account.

- **Three Amigos Discussion:**
  - BA: "What happens if the password is wrong?"
  - Dev: "We’ll show an error message."
  - QA: "Let’s define a scenario for successful login and one for failed login."

---

4. Cucumber

- A popular **BDD framework** that allows writing **executable specifications** in Gherkin.
- Works with multiple programming languages (Java, Python, JavaScript, Ruby, etc.).

---

5. Gherkin Language

Introduction

- **Domain-Specific Language (DSL)** for BDD.
- Uses **plain English** with specific keywords:
  - `Feature`, `Scenario`, `Given`, `When`, `Then`, `And`, `But`, `Background`.

Feature File Example

```
Feature: Login functionality
  In order to access my account
  As a registered user
  I want to be able to log in successfully
```

---

6. Feature Files

- Contain **high-level descriptions** of system features.
- Each file usually represents **one feature**.

Example – Login.feature

```
Feature: Login functionality

  Scenario: Successful login with valid credentials
    Given User is on the login page
    When User enters valid username and password
    And clicks on login button
    Then User should be redirected to the homepage
```

---

7. Scenarios

- Define **specific behavior** of the system.
- Each **Scenario** represents a **test case**.

Example:

```
Scenario: Login fails with incorrect password
  Given User is on the login page
  When User enters valid username and invalid password
  And clicks on login button
  Then User should see an error message "Invalid credentials"
```

---

8. Step Definitions

- Map Gherkin steps to **actual code**.
- Implemented in programming language (e.g., Java, Python).

Example (Java + Selenium):

```
@Given("User is on the login page")
public void user_is_on_login_page() {
    driver.get("https://example.com/login");
}

@When("User enters valid username and password")
public void enter_valid_credentials() {
    driver.findElement(By.id("username")).sendKeys("testUser");
    driver.findElement(By.id("password")).sendKeys("testPass");
}

@Then("User should be redirected to the homepage")
public void verify_homepage() {
    Assert.assertTrue(driver.findElement(By.id("welcomeMsg")).isDisplayed());
}
```

---

9. Parameterization

- Avoids duplication by using **placeholders**.

Example:

```
Scenario Outline: Login with multiple credentials
  Given User is on the login page
  When User enters "<username>" and "<password>"
  And clicks on login button
  Then Login should be "<status>"

  Examples:
    | username | password | status   |
    | testUser | test123  | success  |
    | john     | wrong123 | failure  |
```

---

10. Cucumber Hooks and Tags, Background

Hooks

- `@Before` – Runs before each scenario.
- `@After` – Runs after each scenario.

Example:

```
@Before
public void setup() {
   driver = new ChromeDriver();
}

@After
public void tearDown() {
   driver.quit();
}
```

Tags

- Used to group scenarios.
- Can run a **subset of tests**.

Example:

```
@SmokeTest
Scenario: Successful login
  Given User is on the login page
  When User enters valid username and password
  Then User should be redirected to the homepage
```

Background

- Common steps across scenarios.

```
Feature: Login functionality
  Background:
    Given User is on the login page

  Scenario: Successful login
    When User enters valid credentials
    Then User should be redirected to the homepage

  Scenario: Invalid login
    When User enters invalid credentials
    Then User should see error message
```

---

11. Cucumber Framework in Action

Project Structure:

```
src/test/java
  ├── features
  │     └── Login.feature
  ├── stepDefinitions
  │     └── LoginSteps.java
  ├── runners
  │     └── TestRunner.java
```

Runner File Example (JUnit):

```
@RunWith(Cucumber.class)
@CucumberOptions(
  features = "src/test/java/features",
  glue = {"stepDefinitions"},
  tags = "@SmokeTest",
  plugin = {"pretty", "html:target/cucumber-reports.html"}
)
public class TestRunner {}
```

---

12. Best Practices of Writing Effective Gherkin Language

□ Use **clear, concise language** (avoid technical jargon).
□ Write **independent scenarios** (avoid dependencies).
□ Keep **steps reusable**.
□ Use **Scenario Outline** for multiple data sets.
□ Avoid writing too many steps (5–7 per scenario is ideal).
□ Use **Background** only for common setup.
□ Tag scenarios appropriately (`@Regression`, `@Smoke`).
□ Involve **business + QA + Dev** in scenario writing.

□ Real Scenario (E-commerce Checkout Example)

Feature: Checkout process

```
Feature: Checkout process
  In order to complete my purchase
  As a registered user
  I want to be able to checkout successfully

  Background:
    Given User is logged in
    And User has items in cart

  @SmokeTest
  Scenario: Successful checkout
    When User navigates to checkout page
    And enters valid shipping details
    And makes payment with valid card
    Then User should see order confirmation

  @Regression
  Scenario Outline: Checkout with different payment methods
    When User makes payment using "<paymentMethod>"
    Then User should see "<result>"

    Examples:
      | paymentMethod | result             |
      | Credit Card   | Order confirmation |
      | PayPal        | Order confirmation |
      | Expired Card  | Payment declined   |
```