Lab Title: TDD vs BDD

Lab Number: 2

Objective

The aim of this lab is to understand the difference between Test-Driven Development (TDD) and Behavior-Driven Development (BDD). Both are modern testing approaches in Agile software development, and this lab helps us see how they are applied and how they differ.

Theory

TDD and BDD are both software development techniques that use testing to improve code quality. However, their approach and focus are slightly different.

TDD (Test-Driven Development)

TDD is a process where developers write **tests before writing actual code**. It follows a cycle called **Red → Green → Refactor**:

- Write a test (it fails – Red),
- Write just enough code to pass (it passes – Green),
- Improve the code (Refactor).

This helps developers ensure that every piece of code is tested from the start. It mainly focuses on **how the code works**.

BDD (Behavior-Driven Development)

BDD is an extension of TDD that focuses on **what the system should do** rather than how it does it. It's written in plain language (like English), making it easier for non-developers like testers, clients, or business people to understand. BDD follows a **Given–When–Then** format:

- Given some context,
- When an action occurs,
- Then an outcome should happen.

It focuses more on **how the software behaves** from the user's point of view.

Implementation

Here's how TDD and BDD typically work:

TDD Example (Python):

```
def add(a, b):
    return a + b
```

```
# Test
def test_add():
    assert add(2, 3) == 5
```

BDD Example:

```
Feature: Addition
  Scenario: Add two numbers
    Given I have numbers 2 and 3
    When I add them
    Then the result should be 5
```

BDD tools like **Cucumber** or **Behave** allow writing these behaviors that anyone can read and understand.

Result

From this lab, we understood that:

- TDD is good for developers to write reliable, tested code.
- BDD is more focused on how the system should behave for the user.
- TDD is code-focused, while BDD is behavior-focused.
- BDD encourages better communication between developers and non-developers.

Conclusion

Both TDD and BDD are powerful testing strategies. TDD ensures your code is correct, while BDD ensures your system behaves the way users expect. In Agile projects, using both together can lead to better collaboration, better code, and better products.