**DAY 2 ASSIGNMENT - 5**

**Comparative Infographic: TDD vs BDD vs FDD**

| **Methodology** | **Full Form** | **Focus** |
| --- | --- | --- |
| **TDD** | Test-Driven Development | Code correctness through unit tests |
| **BDD** | Behaviour-Driven Development | Software behaviour from a user perspective |
| **FDD** | Feature-Driven Development | Building software based on features (functionality) |

**1. TDD – Test-Driven Development**

**What is it?**

TDD means you **write the test first**, then you write the actual code.

**Imagine this:**

You are writing a calculator app. Before you write the code to add two numbers, you **write a test** that says “if I add 2 and 3, the answer should be 5.”  
But the code does not exist yet — so the test fails.

Then you write the code to **make the test pass**.

After it passes, you can improve your code if needed (this is called **refactoring**).

**2. BDD – Behaviour-Driven Development**

**What is it?**

BDD is all about how the **user expects the software to behave**. Instead of just writing code, you describe the behaviour in **simple language** like a story.

**Imagine this:**

You are building a login system. You write something like:

**Given** the user is on the login page  
**When** they enter the right username and password  
**Then** they should be logged in

This becomes the base for testing and writing code.  
It helps **developers, testers, and clients** understand what is being built.

**3. FDD – Feature-Driven Development**

**What is it?**

FDD focuses on building **small features one by one**. Each feature is like a mini goal, such as “Allow users to upload a profile picture.”

**Imagine this:**

You are building an e-commerce website. You break it into small features:

* Add to cart
* Checkout
* Payment
* Order tracking

Each feature is planned, designed, coded, and tested **separately**