**Day 4 -Assignment**

* **Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.**

Test-Driven Development (TDD) Process



*Create Precise Tests*

**Write precise tests for the functionality you want to add.**

*Example: “Write a test that checks if the user can log in.”*



Correcting the Code

**Write the minimum amount of code required to pass the test.**

*Example: “Implement the login functionality to pass the test.”*



Refactor the Code

**Refactor the code while ensuring tests still pass, improving code quality.**

*Example: “Clean up the login function, improve readability.”*



**Repeat**

**Repeat the cycle for the next piece of functionality.**

*Example: “Proceed to the next feature or bug fix.”*

Benefits of TDD



Higher Code Quality

**Ensures code is tested thoroughly from the beginning.**



Faster Debugging

**Tests catch bugs early, making them easier to fix.**



Better Design

**Encourages simpler, more modular code.**



Improved Documentation

**Tests serve as documentation for how the code should behave.**

**2.Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.**

**Comparative Analysis of Software Development Methodologies**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test-Driven Development (TDD)** | **Behavior-Driven Development (BDD)** | **Feature-Driven Development (FDD)** |
| **Approach** | Writing Tests First | Focus on Behavior | Feature-Centric |
|  | Iterative Process | Collaborative Process | Five Phases: Develop Overall Model, Build Features by Feature, Plan by Feature, Design by Feature, Build by Feature |
| **Beneﬁts** | Higher Code Quality | Improved Communication | Scalability |
|  | Faster Debugging | Clearer Understanding | Predictability |
| **Suitability** | Best for: Small to medium-sized projects | Best for: Projects with complex business logic and requirements | Best for: Large projects with multiple teams and complex feature sets |
|  | Programming Languages: Suitable for most languages | Domain-Speciﬁc Languages: Uses domain-speciﬁc languages | Structured Approach: Requires a structured approach to project management |
|  | Team Size: Works well with small to medium-sized teams | Team Collaboration: Requires strong collaboration between | Team Coordination: Works well with larger teams and multiple development streams |