Day 4 Assignment

Name: Mehul Anjikhane Email: mehulanjikhane13@gmail.com

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.

Ans:

The TDD Cycle

Test-Driven Development (TDD) is a software development approach that emphasizes writing tests before writing code. Imagine building a house; TDD ensures a solid foundation by first defining what the house should be before construction begins.

Test-Driven Development (TDD) Process Infographic

1. Write Test:
• Developers write automated tests for a small piece of functionality before writing the corresponding production code.
• Tests are written to define the desired behavior and functionality of the code.

2. Run Test:
• Automated test suite is executed to validate the code.• Initial test will fail as no code has been written yet.

3. Write Code:
• Developers write the minimum amount of code necessary to make the failing test pass.
• Focus is on writing only what's needed to fulfill the test requirements.

4. Run Test Again:
• Automated test suite is executed again to verify that the newly written code passes the test.
• Test should now pass, indicating successful implementation of the functionality.

5. Refactor Code:
• Developers refactor the code to improve readability, maintainability, and performance.
• Refactoring is done without changing the behavior of the code as verified by the tests.

Benefits of TDD

- **Reduced Bugs:** Tests act as a safety net, catching errors early in the development process.
- **Improved Design:** Focusing on testable code leads to a more modular and maintainable codebase.
- **Clearer Requirements:** Defining tests clarifies the expected behavior of the software.
- **Increased Confidence:** Passing tests provide a sense of security that the code is workingas intended.
- **Software Reliability:** TDD promotes a disciplined development approach, leading to morereliable software.

How TDD Fosters Software Reliability:

- Continuous Testing: TDD promotes a culture of continuous testing, where everycode change is validated against a suite of automated tests.

- Regression Prevention: Automated tests act as a safety net, catching regressions andensuring that existing functionality remains intact.

- Early Feedback: TDD provides immediate feedback on the correctness of code,allowing developers to quickly identify and fix issues.

Assignment 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.

Ans:

Aspects	**Test-Driven Development (TDD)**	**Behaviour-Driven Development (BDD)**	**Feature-Driven Development (FDD)**
Approach	•Write tests before writing code. •Focuses on writing small, incremental tests to drive the development process.	•Focuses on behavior and outcomes rather than implementation details. •Uses natural language specifications (e.g., Given-When-Then) to define tests.	•Iterative development based on features and domain modelling. •Focuses on building features incrementally based on client priorities.
Benefits	• Early bug detection. • Improved code quality. • Incremental and Iterative development.	• Improved collaboration between stakeholders. • Clear communication of requirements using Given-When-Then. • Encourages user-centric development.	• Scalability for large projects. • Clear feature-based progress tracking. • Efficient resource management.
Suitability	• Ideal for small to medium projects. • Projects requiring high test coverage and reliability	• Projects with complex business logic. • Collaboration-heavy projects involving multiple stakeholders.	• Best for large-scale enterprise projects. • Projects with well defined and stable requirements.

Choosing the Right Approach

- Project size and complexity
- Requirement stability
- Team experience and skill set
- Importance of user experience