**Assignment-5**

**Write TDD, BDD and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts.Use visuals to enhance understanding.**

**Test-Driven Development (TDD):** It is a developer-centric methodology where tests are written before the actual code. The process follows a “Red-Green-Refactor” cycle—first, a failing test is created (red), then code is written to pass the test (green), followed by refactoring the code while ensuring all tests still pass. This approach leads to cleaner, more reliable code and significantly reduces the number of bugs during development. TDD is best suited for projects that require high code quality, such as backend systems, libraries, or APIs.

Test-Driven Development (TDD):

* Focus:

Developer-centric, emphasizing test-first coding to improve code quality and reliability.

* Process:

Follows the "Red-Green-Refactor" cycle: write a failing test (red), write code to make it pass (green), then refactor the code while ensuring all tests still pass.

* Best for:

Projects requiring high code quality, such as backend systems, libraries, or APIs.

**Behavior-Driven Development (BDD):** It expands on TDD by focusing on the system's behavior from the user’s perspective. It uses human-readable language (often with tools like Cucumber or Gherkin) to define scenarios that describe how the software should behave in various situations. This encourages close collaboration between developers, QA testers, and non-technical stakeholders. The clarity in requirements and the shared understanding of behavior make BDD ideal for web applications or customer-facing platforms where user experience is key.

Behavior-Driven Development (BDD):

* Focus: User-centric, capturing system behavior from a user's perspective.
* Process: Uses human-readable language (like Cucumber or Gherkin) to define scenarios that describe how the software should behave.
* Best for: Web applications or customer-facing platforms where user experience is key.

**Feature-Driven Development (FDD):** On the other hand, is a more structured, model-driven methodology that emphasizes the planning and delivery of features in short, iterative cycles. It starts with a comprehensive domain model and a prioritized feature list. Each feature is developed and tested within a few days, promoting rapid progress while maintaining control over scope and quality. FDD is especially effective for large teams or enterprise-scale applications where predictable, scalable development is critical.

Feature-Driven Development (FDD):

* Focus:

Structured, model-driven, emphasizing feature-based development in short, iterative cycles.

* Process:

Starts with a domain model and a prioritized feature list. Each feature is developed and tested in short cycles (e.g., a few days).

* Best for:

Large teams or enterprise-scale applications where predictable, scalable development is critical.