Assignment-2

1. Test-Driven Development (TDD):

Approach: Write tests before writing code. Red-Green-Refactor cycle: Write a failing test (Red), make it pass (Green), refactor code (Refactor).

Benefits: Early bug detection, improved code quality, modular design, faster development cycles.

Suitability: Ideal for Agile environments, small teams, projects with clear requirements, and when unit testing is crucial.

2. Behavior-Driven Development (BDD):

Approach: Focuses on behavior rather than implementation details. Uses a ubiquitous language to describe system behavior in scenarios.

Benefits: Improved collaboration between stakeholders, clearer understanding of requirements, automated acceptance testing.

Suitability: Well-suited for complex systems, projects with changing requirements, large teams, and fostering communication between technical and non-technical stakeholders.

3. Feature-Driven Development (FDD):

Approach: Breaks down development into feature increments. Emphasizes domain object modeling, design by feature, and iterative development.

Benefits: Scalability, flexibility, emphasis on domain modeling, clear visibility of progress, efficient resource allocation.

Suitability: Suitable for large-scale projects, teams with varying skill levels, projects requiring rapid development and delivery, and when a structured process is needed.