Assignment 02

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.

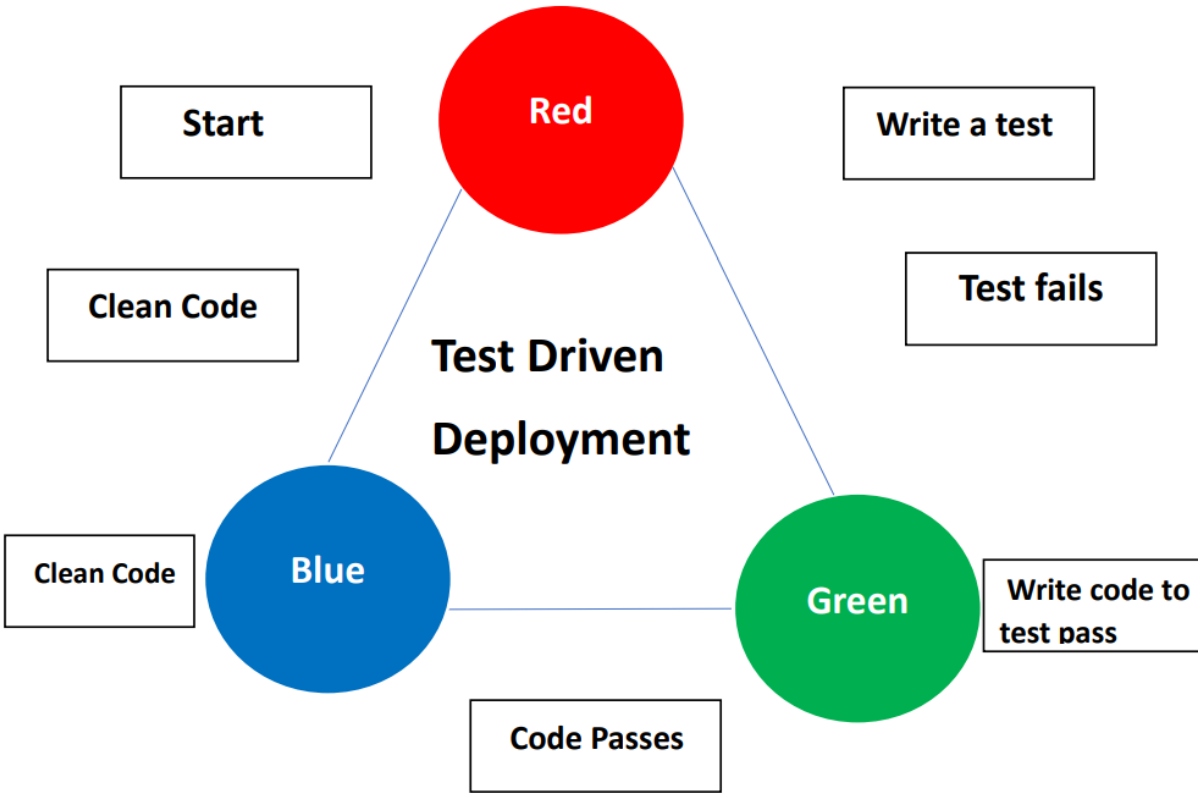
Test-Driven Development (TDD)

1. Approach

   . Write tests before writing code.

   . Focus on small, incremental development cycles.

   . Tests define the desired behaviour of the code.



2. Benefits

. Early bug detection and reduction.

. Improved code quality and design.

. Enhanced software reliability through continuous testing.

3. Suitability

. Ideal for Agile environments.

. Well-suited for projects with clear requirements.

. Best for projects where code quality is paramount.

Behavior-Driven Development (BDD)

1. Approach

. Focuses on behavior rather than implementation details.

. Uses natural language specifications (e.g., Gherkin syntax).

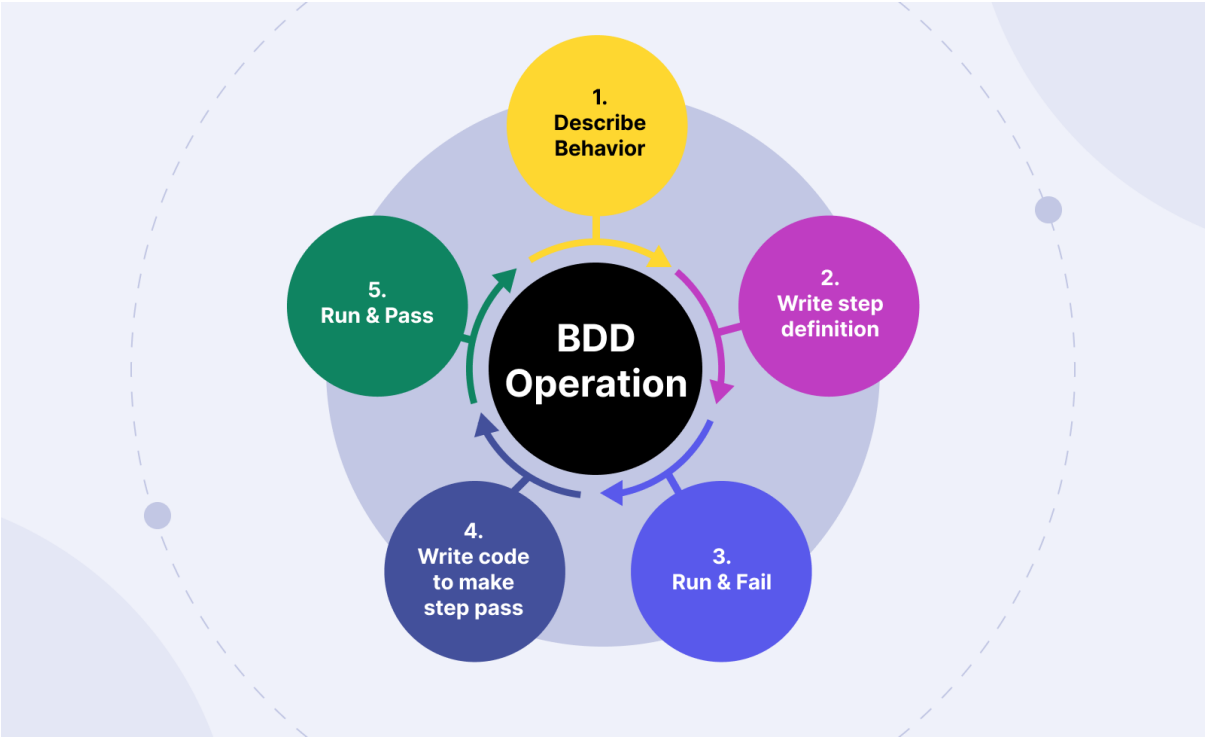
. Collaboration between developers, QA, and stakeholders.

2. Benefits

. Encourages clear communication between stakeholders.

. Promotes a shared understanding of requirements.

. Facilitates automated testing and documentation.



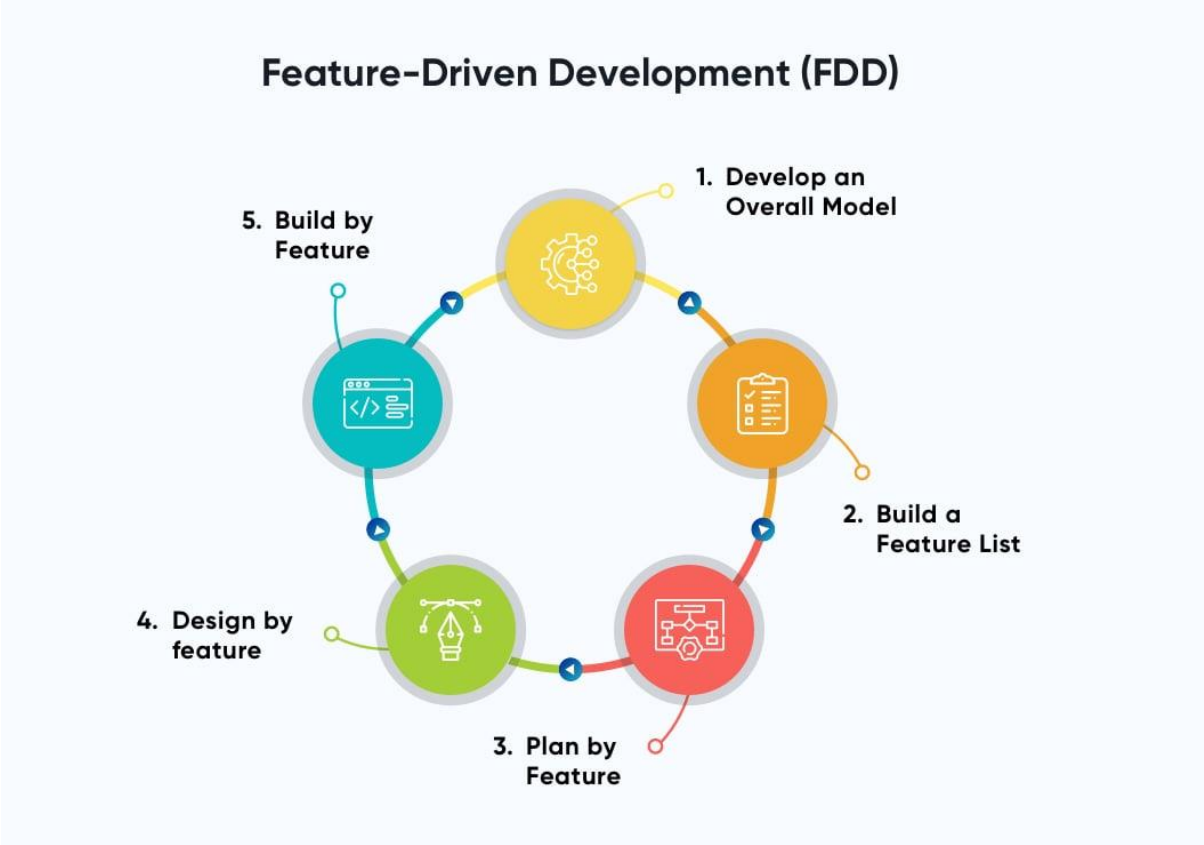
3. Suitability

- Useful for projects with complex business logic.
- Suitable for teams with diverse skill sets.
- Effective for projects requiring a high level of stakeholder involvement.

Feature-Driven Development (FDD)

1. Approach

- Iterative and incremental development based on features.
- Features are broken down into smaller tasks and implemented sequentially.
- Emphasizes domain modeling and feature ownership.



2.Benefits

- Clear focus on delivering tangible features.
- Effective for large-scale projects with multiple teams.
- Encourages frequent releases and feedback loops.

3.Suitability

- Suitable for large, complex projects with changing requirements.
- Ideal for teams with a strong emphasis on domain modeling.
- Works well in environments where feature delivery is critical.