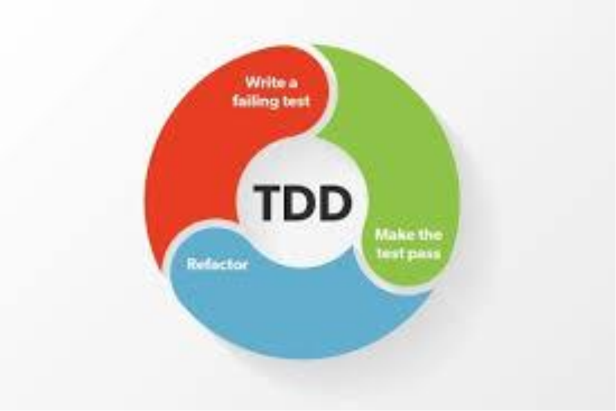
Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

Test-Driven Development (TDD)



Approach:

- Developers write test cases before writing the code.
- Code is iteratively developed and refined to pass the tests.

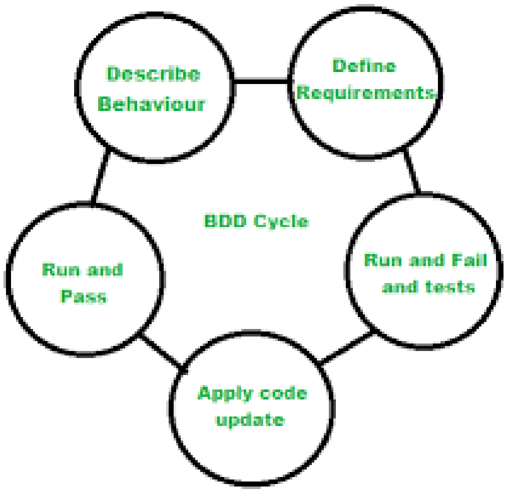
Benefits:

- Ensures code functionality as per requirements.
- Encourages modular and well-structured code.
- Early detection of bugs, leading to lower debugging costs.
- Facilitates continuous integration and deployment.

Suitability:

- Ideal for Agile and iterative development.
- Suitable for projects with clear, well-defined requirements.
- Effective for small to medium-sized projects.

Behavior-Driven Development (BDD)



Approach:

- Focuses on defining behavior through examples written in a human-readable format (e.g., Gherkin syntax).
- Tests are written from a user's perspective.
- Emphasizes collaboration between developers, testers, and stakeholders.

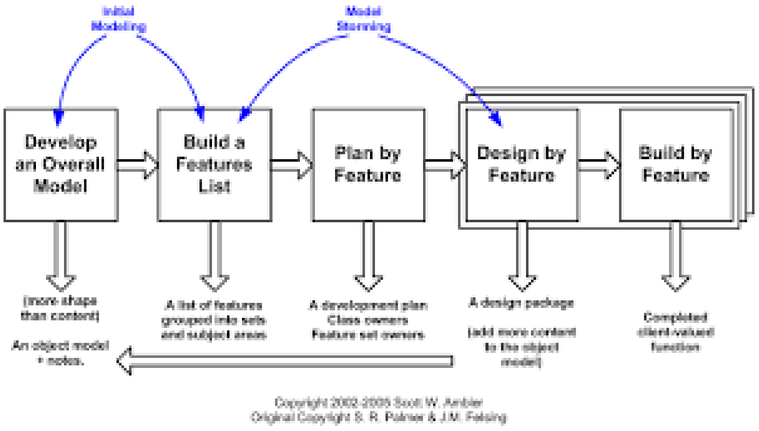
Benefits:

- Promotes a shared understanding of requirements among stakeholders.
- Improves communication and collaboration between teams.
- Encourages test automation and documentation.
- Helps in identifying acceptance criteria early in the development process.

Suitability:

- Particularly useful for projects with complex business logic.
- Suitable for projects with diverse stakeholders and team members.
- Effective for large-scale projects with evolving requirements.

Feature-Driven Development (FDD)


Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan
Class owners
Feature set owners
A design package
(add more content to the object model)
Completed client-valued function
Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

Approach:

- Breaks down the development process into small, feature-specific tasks.
- Emphasizes iterative and incremental development.
- Focuses on domain object modeling and design inspection.

Benefits:

- Offers a systematic approach to feature prioritization and development.
- Promotes rapid delivery of tangible, working features.
- Encourages collaboration and communication within the development team.
- Suitable for large and complex projects with multiple development teams.

Suitability:

- Well-suited for projects with changing requirements and evolving feature sets.
- Effective for large-scale enterprise software development.
- Ideal for projects requiring a structured and disciplined development process.