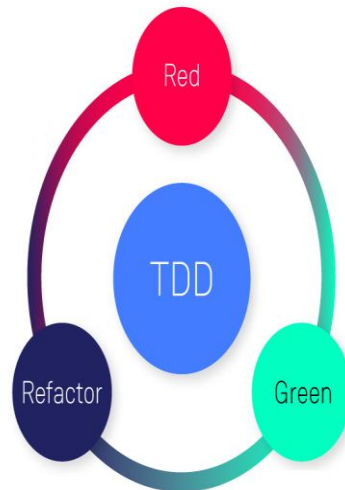


Assignment 2:

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:

TDD follows a "test-first" approach, where developers write tests before writing the actual code. The tests guide the development process, and the code is written to pass those tests.

- Benefits:

TDD helps identify and fix bugs early.

Improves software reliability.

Provides clearer requirements.

Offers a faster feedback loop.

Promotes maintainable code

Facilitates regression testing.

Suitability:

TDD is suitable for projects where the requirements are well-defined and can be broken down into smaller units. It works well for projects with frequent changes and a focus on code quality and reliability.

Behavior-Driven Development (BDD):



Approach:

BDD focuses on the behavior of the software from a user's perspective. It emphasizes collaboration between developers, testers, and business stakeholders to define and validate the expected behavior using a common language.

Benefits:

BDD improves communication and collaboration.

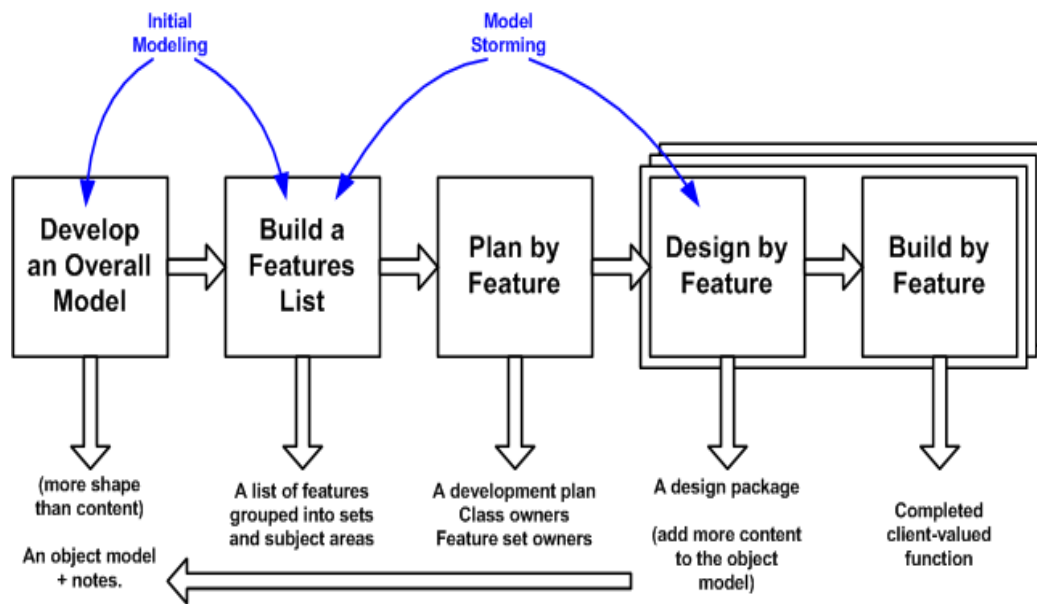
Enhances the understanding of requirements.

Promotes test coverage, provides living documentation, and facilitates the creation of acceptance tests.

Suitability:

BDD is suitable for projects with complex requirements, multiple stakeholders, and a need for clear and shared understanding of the software's behavior. It is particularly useful for projects with a strong focus on user experience and behavior.

Feature-Driven Development (FDD):



Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

Approach:

FDD is an iterative and incremental software development approach that focuses on delivering features in short iterations. It involves a five-step process: domain walkthrough, design by feature, build by feature, code inspection, and feature acceptance.

Benefits:

FDD promotes a systematic and structured approach to development.

Improves team collaboration.

Provides clear visibility of progress.

Allows for parallel development, and emphasizes on delivering working software incrementally.

Suitability:

FDD is suitable for large-scale projects with a well-defined scope and a focus on delivering features. It works well for projects with a stable and experienced development team and a need for clear progress tracking.

Comparison –

Feature	TDD	BDD	FDD
Focus	Code-centric	Behavior-centric	Feature-centric
Test Style	Unit test	Acceptance test	Integration test
Collaboration	Limited	Extensive	Moderate
Documentation	Minimal	Detailed	Moderate
Complexity	Low to Medium	Medium	High
Flexibility	Low	Medium	High
Adaptability	High	High	High

Conclusion –

Each methodology has its unique approach, benefits, and suitability for different software development contexts. It's important to consider the project's requirements, team dynamics, and the desired outcomes when choosing the most appropriate methodology.