

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.

This infographic compares three popular approaches: Test-Driven Development (TDD), Behavior-Driven Development (BDD), and Feature-Driven Development (FDD).

TDD, BDD, & FDD Methodologies



TDD

Test - Driven Development

BDD

Behaviour - Driven Development

FDD

Feature - Driven Development

Approach –

TDD (Test-Driven Development)

- ✓ **Focus** – Writing unit tests before writing code.
- ✓ **Process** –
 - Define a test for a specific functionality.
 - Write minimal code to make the test pass.
 - Refactor the code for maintainability.
- ✓ **Visual** – A magnifying glass inspecting code lines.

BDD (Behavior-Driven Development)

- ✓ **Focus** – Collaboration between developers, testers, and stakeholders on defining software behavior.
- ✓ **Process** –
 - Develop user stories outlining desired functionalities.
 - Translate user stories into executable specifications (acceptance criteria).
 - Develop and test the software based on acceptance criteria.
- ✓ **Visual** – A group of people sitting around a table discussing ideas.

FDD (Feature-Driven Development)

- ✓ **Focus** – Breaking down large projects into smaller, manageable features.
- ✓ **Process** –
 - Develop a feature list with clear definitions and priorities.
 - Divide features into iterative development cycles.
 - Design, develop, and test features within each cycle.
- ✓ **Visual** – A puzzle being assembled piece by piece.

Benefits –

- ✓ **TDD**
 - Reduces bugs
 - Improves code design
 - Encourages clean code practices
- ✓ **BDD**
 - Improves communication and collaboration
 - Ensures software meets user needs

- Provides clear acceptance criteria
- ✓ **FDD**
 - Simplifies project management
 - Enables early risk identification
 - Delivers working features in predictable increments

Suitability –

- ✓ **TDD:** Ideal for unit testing, improving code quality for smaller, well-defined functionalities.
- ✓ **BDD:** Well-suited for projects with complex user interactions and a need for strong stakeholder involvement.
- ✓ **FDD:** Effective for large-scale projects requiring a structured approach to feature development and delivery.

Conclusion –

TDD, BDD, and FDD offer valuable tools for software development. The best approach depends on your project's specific needs and priorities. Consider the factors discussed in this infographic to make an informed decision about the methodology that best suits your development process.
