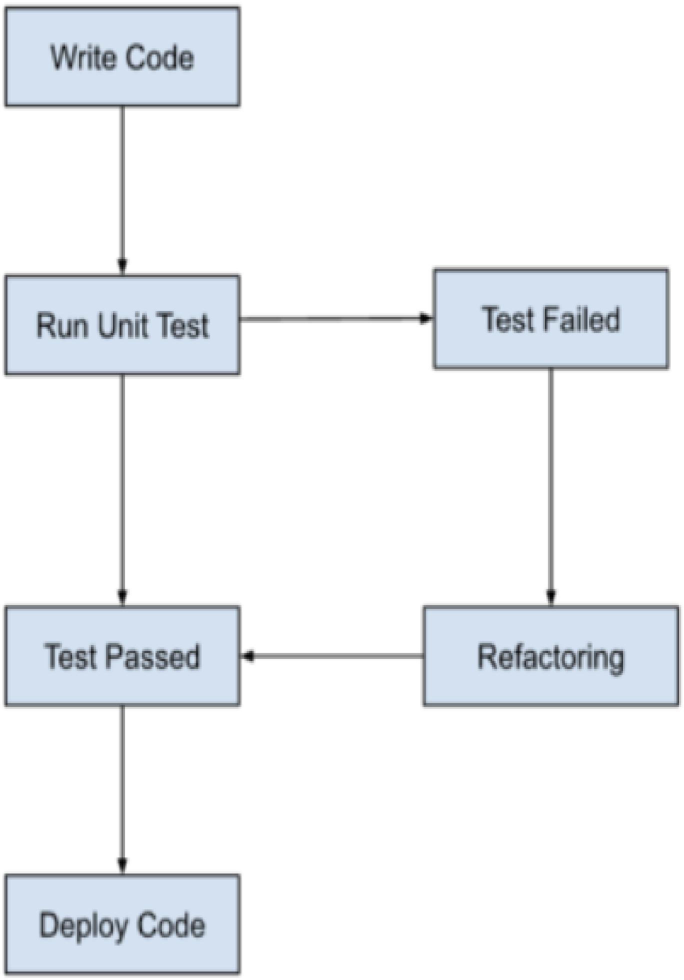
Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

Test-Driven Development (TDD)



Approach:

- Write tests before writing code.
- Focus on small units of code (unit tests).

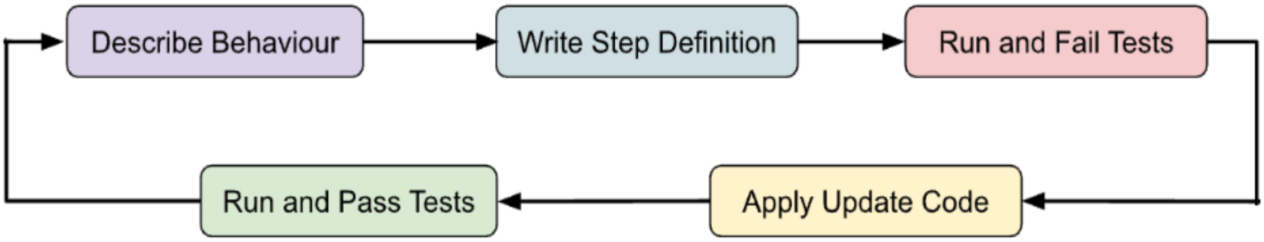
Benefits:

- Early bug detection.
- Incremental development.
- Improved code quality.

Suitability:

- Ideal for projects requiring frequent changes.
- Suitable for teams emphasizing code quality and reliability.

A flowchart showing the steps of BDD



Approach:

- Define behaviour using domain-specific language (DSL).
- Tests are written in a human-readable format.

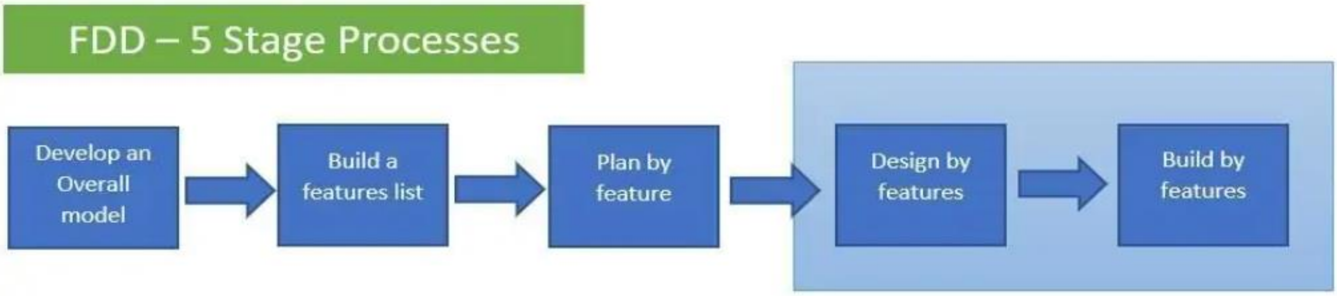
Benefits:

- Enhanced collaboration between stakeholders.
- Clear communication of requirements.
- Focus on business value.
-

Suitability:

- Well-suited for projects with diverse stakeholders.
- Useful for projects emphasizing user behaviour and acceptance criteria.

A flowchart showing the steps of FDD



Approach:

- Divide development into features.
- Iteratively develop features with specific roles and processes.

Benefits:

- Emphasis on feature delivery.
- Clear project progress tracking.
- Scalable for large teams and projects.

Suitability:

- Suitable for large-scale projects with complex requirements.
- Ideal for teams focusing on feature delivery and management.

Conclusion:

- TDD emphasizes writing tests before code, suitable for projects requiring code quality and frequent changes.
- BDD focuses on defining behaviour in a human-readable format, facilitating collaboration and clear communication of requirements.
- FDD divides development into features, ideal for large-scale projects with complex requirements, emphasizing feature delivery and management.