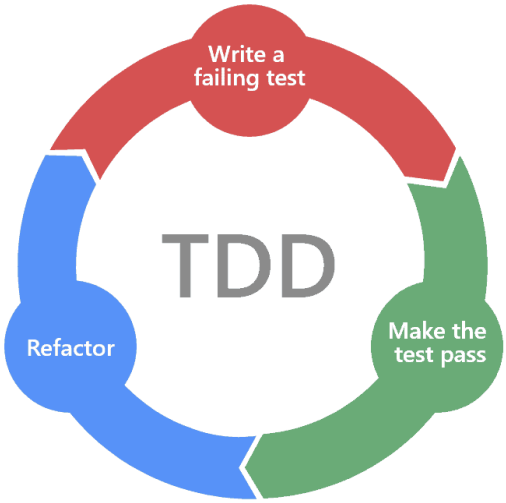
Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

Test-Driven Development (TDD)



- Approach: Write tests before writing code.
- Process:
  1. Write failing test cases.
  2. Implement code to pass the tests.
  3. Refactor the code for better design.
- Icon: Test-first approach.

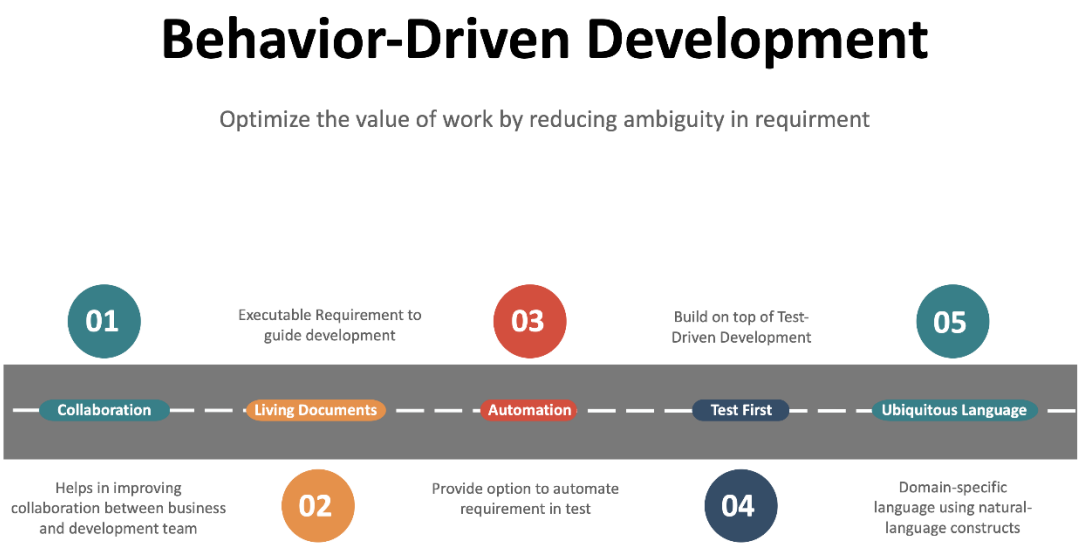
Benefits

- Bug Reduction: Early detection of bugs leads to more robust code.
- Code Quality: Encourages cleaner, modular code.
- Documentation: Tests serve as living documentation.

Suitability

- Ideal For: Projects requiring high reliability (e.g., financial software).
- Context: When quality and bug-free code are critical.

Behavior-Driven Development (BDD)



- Approach: Focus on defining behavior from a user's perspective.
- Process:
  1. Define user stories.
  2. Write scenarios in Given-When-Then format.
  3. Implement code to fulfill scenarios.
- Icon: User stories and behavior focus.

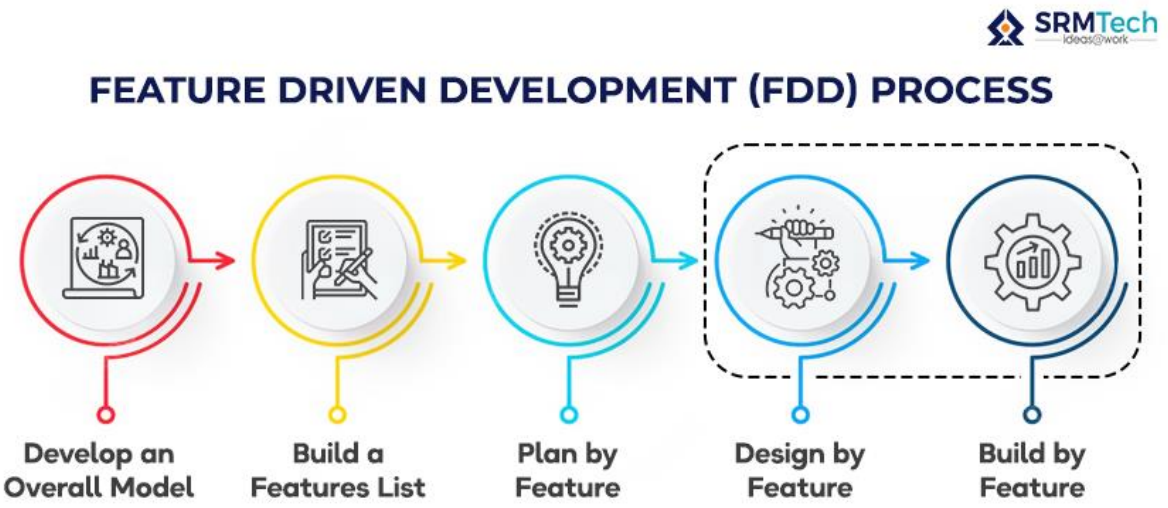
Benefits

- Collaboration: Enhances communication between stakeholders.
- User Focused: Ensures software meets user requirements.
- Clear Requirements: Scenarios clarify feature expectations.

Suitability

- Ideal For: Projects with significant stakeholder involvement (e.g., e-commerce platforms).
- Context: When clear communication and user satisfaction are priorities.

Feature-Driven Development (FDD)



- Approach: Develop features as the primary unit of progress.
- Process:
  1. Develop an overall model.
  2. Build a feature list.
  3. Plan, design, and build features incrementally.
- Icon: Feature-based development.

Benefits

- Scalability: Suitable for large, complex projects.
- Feature Focused: Prioritizes features delivering value.
- Efficiency: Streamlines development with a feature-centric approach.
- Suitability for Different Contexts

Suitability

- Ideal For: Large-scale projects with numerous features (e.g., enterprise systems).
- Context: When feature delivery and scalability are crucial.