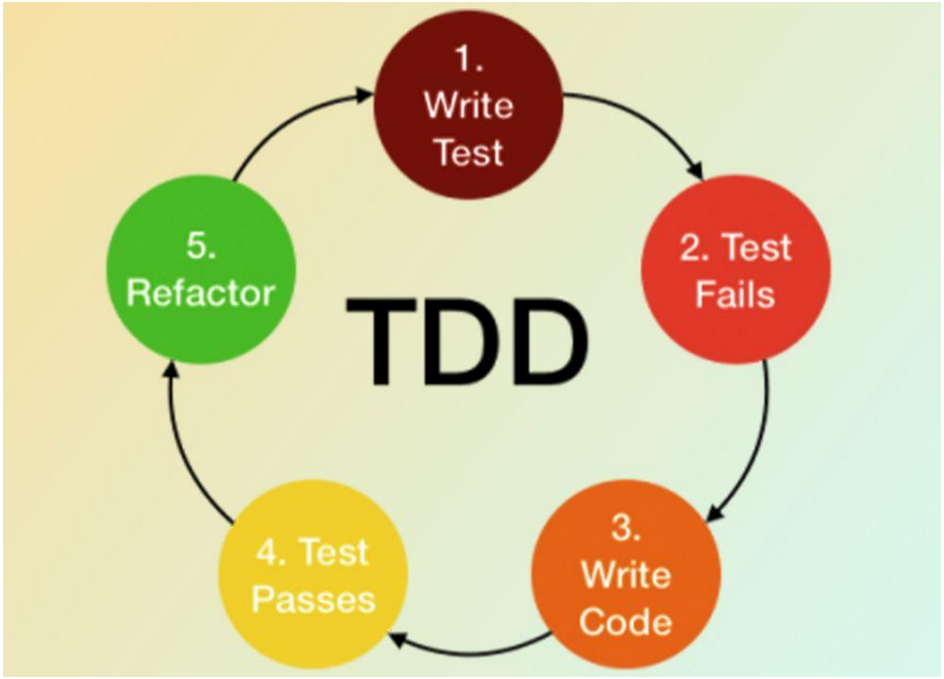
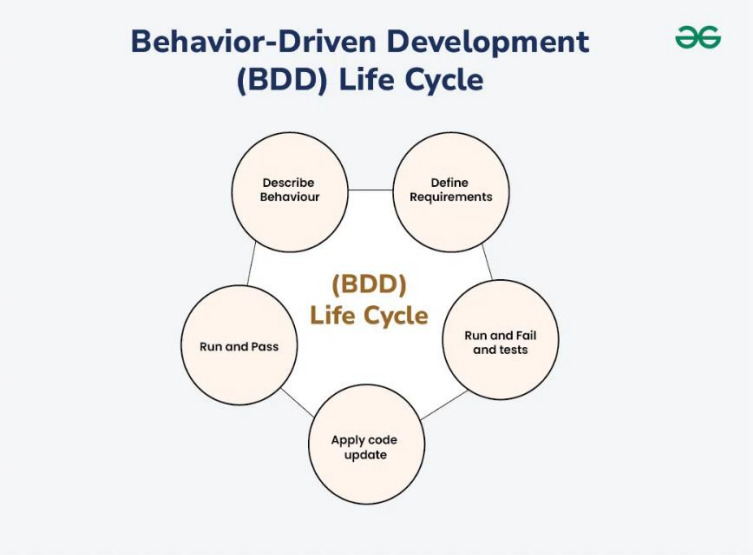
Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.



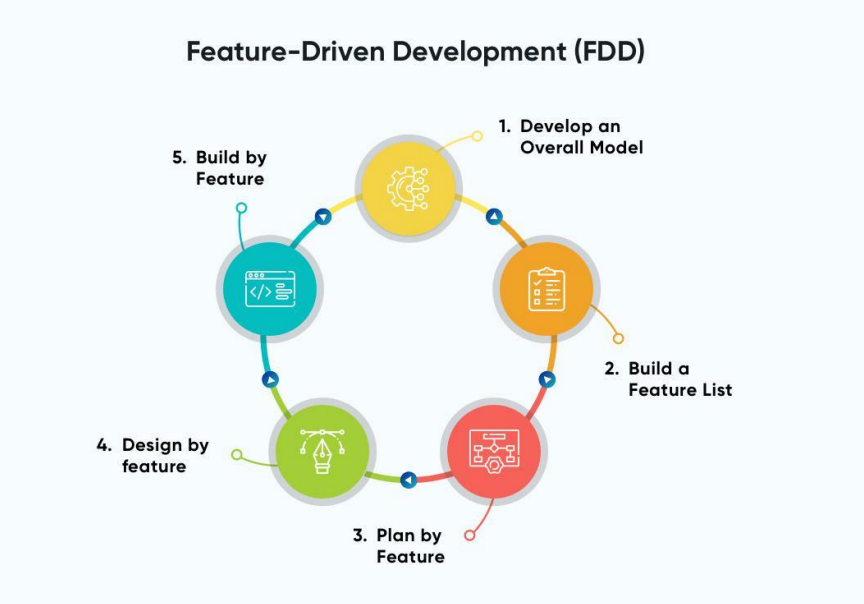
TDD (Test-Driven Development)

- **Approach:**
  - **Write Tests First:** Develop tests based on expected behavior before writing the actual code.
  - **Incremental Development:** Iteratively add small pieces of functionality that pass the written tests.
- **Benefits:**
  - **Early Bug Detection:** Identifies defects early in the development process.
  - **Clean Code:** Encourages modular and maintainable code due to frequent refactoring.
- **Suitability:**
  - **Projects:** Best suited for projects with clearly defined requirements and stable specifications.
  - **Environments:** Ideal for Agile and iterative development environments where flexibility and responsiveness are key.



BDD (Behavior-Driven Development)

- **Approach:**
  - **Behavior Focus:** Focuses on defining behavior and expected outcomes using natural language.
  - **Collaboration:** Involves stakeholders (developers, QA, business analysts) in defining and validating scenarios using Given-When-Then format.
- **Benefits:**
  - **Improved Collaboration:** Enhances communication and collaboration between technical and non-technical stakeholders.
  - **Clarity in Requirements:** Helps ensure that software features align with business goals and user expectations.
- **Suitability:**
  - **Projects:** Suitable for complex projects with intricate business logic and diverse stakeholder inputs.
  - **Teams:** Effective in teams striving for better alignment between development and business goals.



FDD (Feature-Driven Development)

- **Approach:**
  - **Feature-Centric:** Iteratively develops features based on domain object modelling and feature lists.
  - **Phases:** Includes planning, designing, developing, and inspecting features in short cycles.
- **Benefits:**
  - **Progress Tracking:** Provides clear visibility into project progress through feature completion.
  - **Risk Management:** Focuses on delivering tangible, working features, reducing project risks.
- **Suitability:**
  - **Projects:** Best for large-scale projects requiring structured planning and management of features.
  - **Teams:** Suitable for teams with domain-specific expertise aiming for systematic feature delivery.