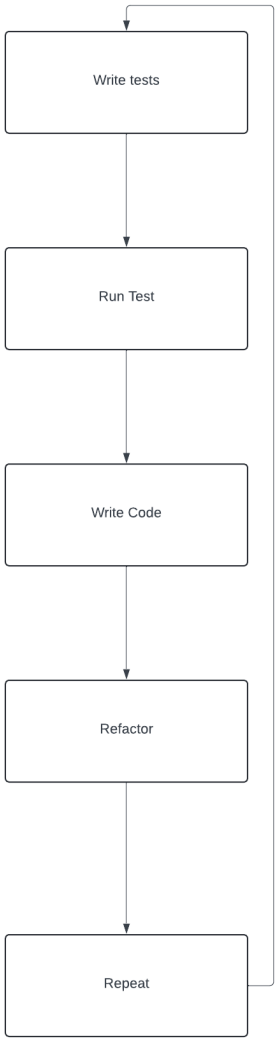
Assignment 1:

Crate an infographic illustrating the Test-Driven Development(TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction and how it fosters software reliability.

Ans:

Below is the infographic illustrating Test-Driven Development(TDD)



Below are the steps from TDD process:

- Write tests:
  Write unit tests before writing any code
  Tests should cover the expected behaviour of the code
- Run tests:
  Execute tests to see them fail

Failures indicate tat the code does not yet satisfy the desired functionality

- Write code:
  Develop the minimum amount of code necessary to pass the failing tests.
  Focus on simplicity and only implement what is required to make tests pass.
- Refactor:
  Refactor the code to improve its design, readability and performance.
  Ensure that all tests continue to pass after refactoring
- Repeat:
  Repeat the process iteratively for each new feature or functionality

Benefits of TDD are:

- Bug reduction:
  Catching bugs early in the development process reduces the likelihood of them reaching production.
- Improved Software Reliability:
  Writing tests before code ensures that the software meets the specified requirements and behaves as expected.
- Faster Development:
  TDD can lead to faster development cycle as developers spend less time debugging and fixing issues later in the process.

Assignment 2:

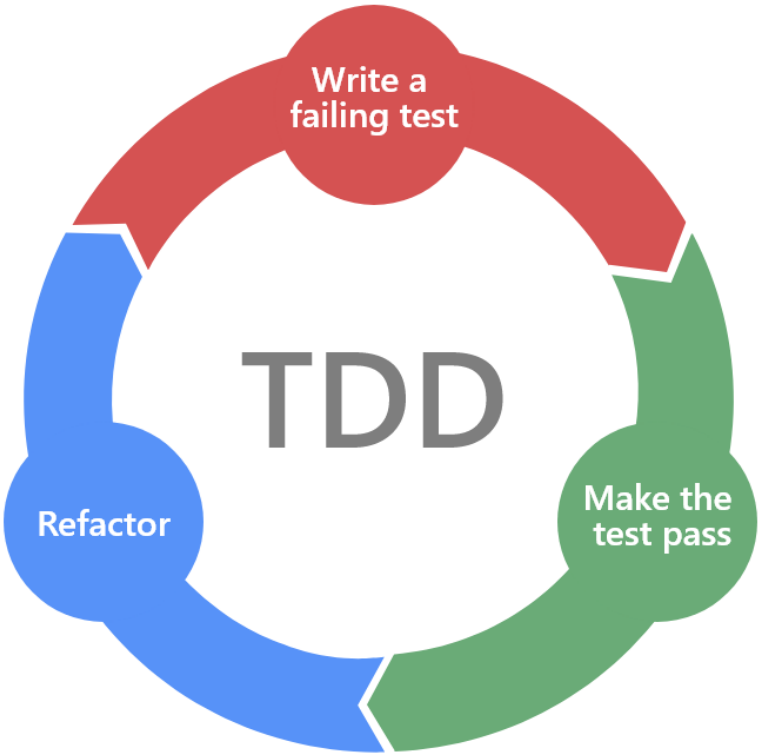
Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

Ans:

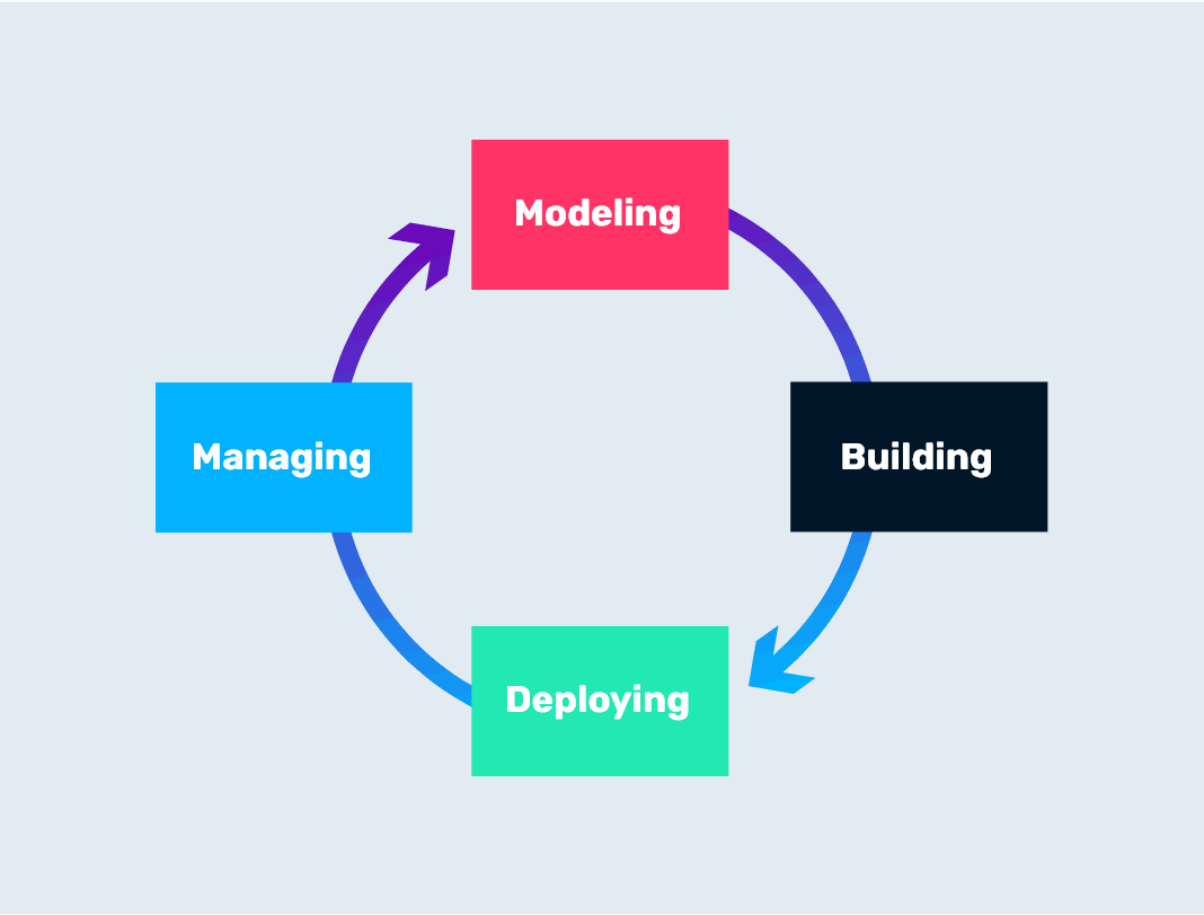
Methodology Comparison TDD vs BDD vs FDD

- Approach:
  - In TDD write tests before coding. Tests drive the development process.
  - In BDD focus on behaviour and collaboration between stakeholders.
  - In FDD develop features incrementally based on client requirements. Each feature is a separate development cycle.
- Testing Focus:
  - Unit testing: Focusses on unit tests for individual components.
  - Acceptance testing: Tests focus on overall system behaviour form the user's perspective.
  - Integration Testing: Emphasis on testing interaction between different components or modules

- Language:
    - Developer centric language.
    - Domain specific language
    - In a language that is descriptive and easy to understand by all.
- Benefits:
    - Bug reduction, enhanced code quality, early detection of issues.
    - Improved communication, focus on user requirements, reduced ambiguity.
    - Client satisfaction, rapid development, clear project scope.
- Suitable for:
    - Agile development, projects with clear requirements.
    - Collaboration-heavy projects. Projects with complex user interactions.
    - Projects with evolving requirements, large scale projects.



BDD


Modeling
Building
Deploying
Managing

FDD


DEVELOP AN OVERALL MODEL
BUILD A FEATURE LIST
PLAN BY FEATURE
DESIGN BY FEATURE
BUILD BY FEATURE