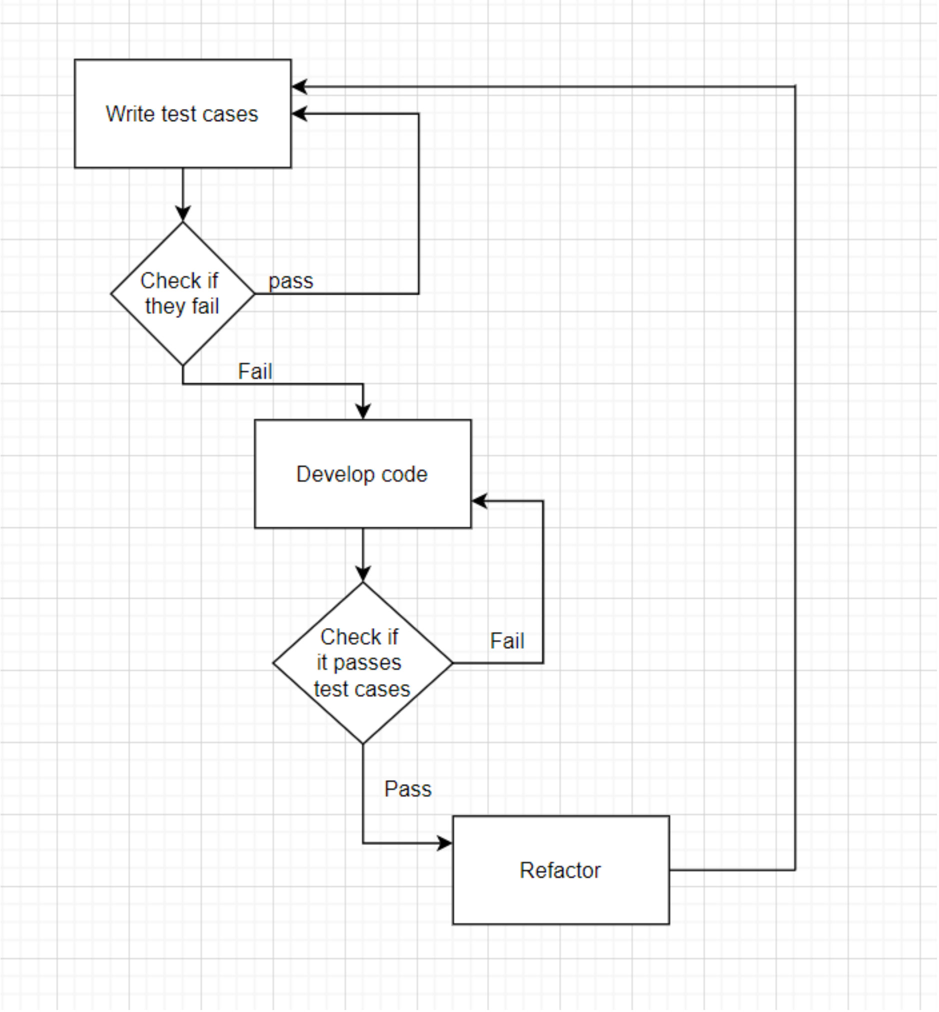
Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.



Test-Driven Development (TDD) Process

1. **Writing Tests Before Code:**
   - In TDD, developers start by writing automated test cases for small units of functionality.
   - Tests are written based on the expected behaviour of the code, focusing on specific requirements.

2. **Running Test Cases:**
   - Once tests are written, they are executed to ensure they fail. This step verifies that the tests are correctly assessing the functionality.

3. **Code Implementation:**
    - Developers then write the minimal amount of code needed to pass the failing test cases.
    - This code should fulfil the requirements specified in the tests, promoting simplicity and clarity.

4. **Refactoring:**
    - After code implementation, developers refactor the code to improve its structure and maintainability.
    - Refactoring ensures that the code remains clean and readable while passing all tests.

5. **Re-running Tests:**
    - Test cases are re-run to verify that the refactored code still passes all tests.
    - This step ensures that changes made during refactoring do not introduce new bugs or regressions.

Benefits of TDD:

1. **Bug Reduction:**
    - By writing tests before code, TDD helps identify and address bugs early in the development process.
    - Catching bugs early reduces the cost and effort required to fix them later in the development cycle.

2. **Improved Code Quality:**
    - TDD encourages developers to focus on writing clean, modular code that fulfills specific requirements.
    - The process promotes software reliability by ensuring that each piece of functionality is thoroughly tested.

3. **Fosters Software Reliability:**
    - TDD fosters software reliability by providing a safety net of automated tests.
    - These tests act as a safeguard against unintended changes or regressions, maintaining the integrity of the codebase.

Assignment3: Produce a comparative infographic of TDD, BDD, and FDD methodologies. IllustrSSate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.