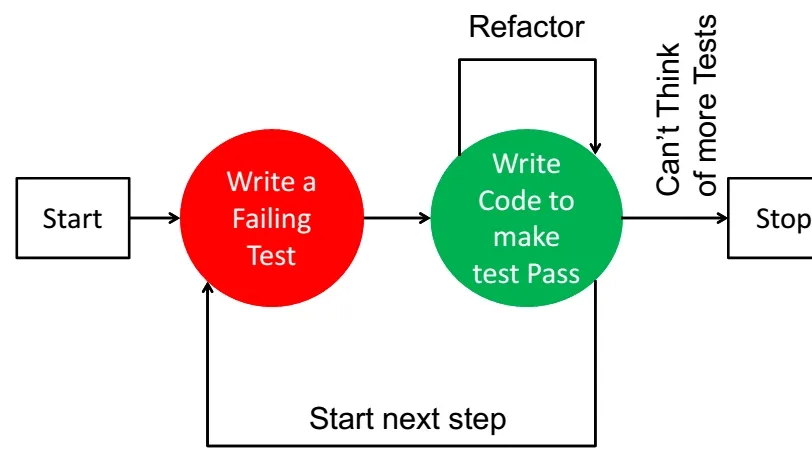
**Day 3 Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.**

**TTD:** -TDD is a software development process where tests are written before the code.

- Ensures that code is tested and meets the requirement

The TDD cycle is then repeated, starting with the creation of a new test for the next piece of functionality. This process helps ensure that your code is always backed by tests, making it easier to catch bugs and regressions as the codebase evolves.

****

**Benefits of TDD:**

**1. Improved Code Quality:** TDD enforces a focus on writing clean, maintainable, and modular code from the outset. By writing tests first, developers must think critically about the design and architecture of their code, leading to higher code quality and fewer design flaws.

**2.** **Reduced Bugs and Defects:** With TDD, bugs and defects are identified early in the development process as tests are written before code implementation. This proactive approach helps catch issues before they propagate and become more challenging and costly to fix.

**3. Faster Debugging and Development:** TDD accelerates the debugging process by pinpointing issues in smaller, isolated sections of code. This leads to quicker identification and resolution of problems, ultimately speeding up the overall development cycle.

**4. Confident Refactoring:** TDD provides the confidence to refactor code without fear of breaking existing functionality. If tests pass after refactoring, developers can be assured that their changes haven’t introduced new defects, resulting in a more maintainable and adaptable codebase.

These benefits make Test-Driven Development a powerful practice for creating high-quality software with fewer defects, faster development cycles, and increased developer confidence.

**Test Driven Development (TDD) Examples:**

**Calculator Function:** When building a calculator function, a TDD approach would involve writing a test case for the “add” function and then writing the code for the process to pass that test. Once the “add” function is working correctly, additional test cases would be written for other functions such as “subtract”, “multiply” and “divide”.

**User Authentication:** When building a user authentication system, a TDD approach would involve writing a test case for the user login functionality and then writing the code for the login process to pass that test. Once the login functionality works correctly, additional test cases will be written for registration, password reset, and account verification.

**E-commerce Website:** When building an e-commerce website, a TDD approach would involve writing test cases for various features such as product listings, shopping cart functionality, and checkout process. Tests would be written to ensure the system works correctly at each process stage, from adding items to the cart to completing the purchase.