

**Test-Driven Development (TDD) Infographic**

**Step 1: Write a Test**

Identify a small unit of functionality.

Write a test that defines the desired behavior.

The test initially fails as the code does not exist yet.

**Step 2: Write the Code**

Implement the code that fulfills the requirements of the test.

The aim is to make the test pass, no more, no less.

**Step 3: Run Tests**

Execute all tests to ensure the new code meets expectations.

All tests should pass, indicating successful implementation.

**Benefits of TDD:**

Bug Reduction: By writing tests first, developers catch and fix bugs early in the development process.

Improved Design: TDD encourages modular, loosely coupled code, leading to better software architecture.

Documentation: Tests serve as executable documentation, helping new developers understand code functionality.

Confidence in Refactoring: Developers can refactor code confidently, knowing that tests will detect any regressions.

Reliability: TDD fosters software reliability by ensuring that each piece of code has been thoroughly tested.

**Conclusion:**

Test-Driven Development (TDD) is a powerful methodology that prioritizes writing tests before code. By following this process, developers can create more reliable, bug-free software while fostering better design and maintainability.