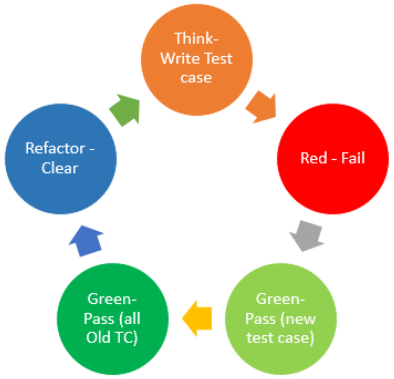
Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.



Test-Driven Development (TDD) is a software development approach where tests are written before the actual code. This ensures that the code is designed to meet specific requirements and helps to catch bugs early in the development process.

Write Test Case: Before writing any code, the process begins with creating test cases for the intended functionality. The expected behavior of the code is defined by these tests.

Run Tests: Tests should then be conducted. As of now, none of the tests should pass because no code has been developed.

Write Code: Write the code necessary to pass the tests that are failing. Writing the simplest possible code to pass the tests is the aim.

Repeatedly Run the Tests: After writing the code, repeat the tests. Now, all exams ought to pass. If any test is unsuccessful, a regression has been introduced by the code changes.

Refactor Code (Optional): Refactor the code to maintain all tests passing while making it more readable, performant, or well-designed.

Benefits of TDD

Reduced Bugs: By writing tests first, TDD helps to identify and fix bugs early in the development process, leading to higher quality software.

Improved Design: The focus on writing tests before code can lead to a more well-designed and maintainable codebase.

Increased Confidence: With a suite of automated tests, developers can have more confidence that their code is working as expected.

Faster Development: While TDD may seem slower initially, it can actually lead to faster development in the long run by reducing the need for rework and debugging.

Enhanced Reliability: TDD helps to ensure that the software is reliable and meets the needs of its users.

TDD: Building Software You Can Trust

By following the TDD approach, developers can create software that is more reliable, maintainable, and less prone to bugs. This leads to a better development experience and a higher quality product for end users.