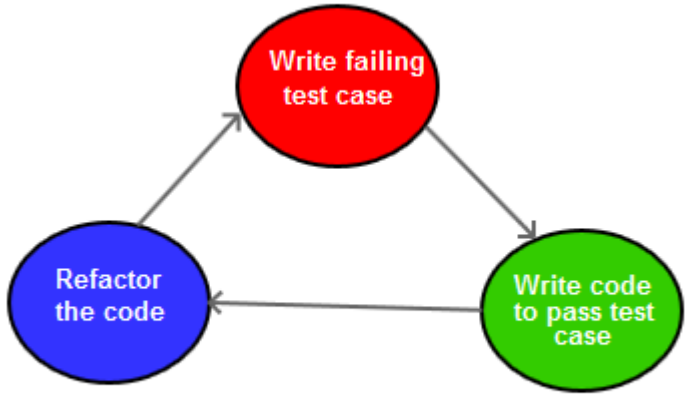
TDD (Test-driven development)

Write code only to fix a failing test case. That is the gist of test driven development (TDD). This is a cyclic process- You first write a test for a requirement, and then you write some real code to pass the test, then you refactor the code for best possible design using various design principle for example SOLID , GRASP etc. The benefit here is that you already have a safety net as tests before refactoring the code. Whatever refactoring you are doing, your tests should not fail.

TDD is an evolving process. You evolve your software design by simply following above steps for each and every requirement of your software. It keeps you away from verbose code, over engineering and making unnecessary assumption.

The ***Test->Code->Refactor*** cycle also known as red-green-blue. You can understand the significance of these colors from below diagram.



We will follow the same cycle

Write a failing test Ỉ make it pass Ỉ refactor the codeȥ

Shopping cart application development using TDD Ỉ shared separatelyȥ

ReferencesȝTest-driven development - Wikipedia

Test Driven Development (TDD) - GeeksforGeeks

ibm.com/garage/method/practices/code/practice_test_driven_development/#:~:text=Test-driven development (TDD),← Back to Develop practices