Test Driven Development

Introduction to TDD

Produced
by:
Dr. Siobhán Drohan
Mairead Meagher
Eamonn de Leastar

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topic List

– What is Test Driven Development?

– What is Unit Testing?

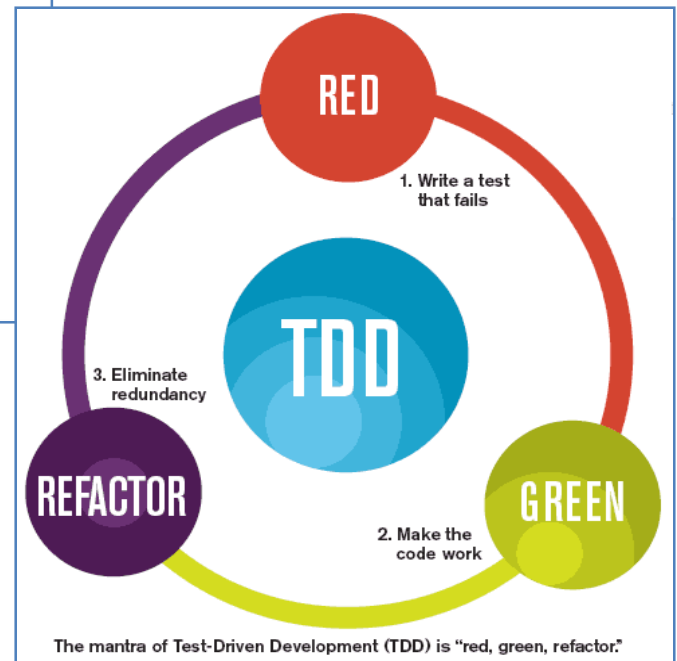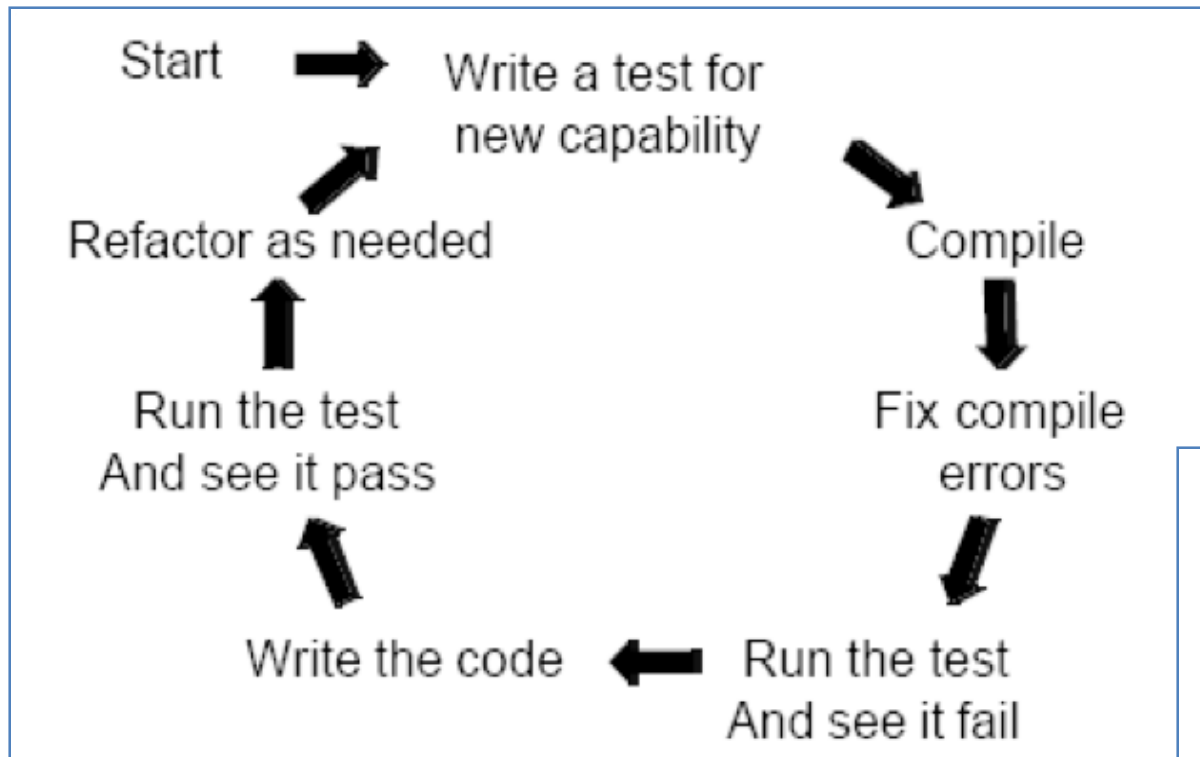– The JUnit Framework.

Test Driven Development

- Tests are written <u>before</u> you write the program code.

- A test framework is used so that <u>automated</u> testing can be done after every small change to the code.

  - This may be as often as every 5 or 10 minutes.

- This approach must be learned and practiced.

Why bother with Test Driven Development?

- It makes your designs better.

- Fewer bugs; drastically reduces the amount of time spent debugging code.

- More maintainable code.

- During development, the program always works—it may not do everything required, but what it does, it does right.

- Your new philosophy (axiom):

  - *'Code that isn't tested doesn't work'*

Test-driven development.

Start ➡ Write a test for new capability

Refactor as needed

Run the test And see it pass

Compile

Fix compile errors

Write the code ⬅ Run the test And see it fail

RED — 1. Write a test that fails

GREEN — 2. Make the code work

REFACTOR — 3. Eliminate redundancy

TDD

The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Topic List

– What is Test Driven Development?

– What is Unit Testing?

– The JUnit Framework.

What is Unit Testing?

- A unit test is a piece of code written by a developer that tests (exercises) a very small, specific area of functionality of the code being tested.

  - Usually a unit test exercises some particular method in a particular context.

What is Unit Testing?

- A unit test is a piece of code written by a developer that tests (exercises) a very small, specific area of functionality of the code being tested.

  – Usually a unit test exercises some particular method in a particular context.

- Unit tests are performed to prove that a piece of code does what the developer thinks it should do.

- The question remains open as to whether that's the right thing to do according to the customer or end-user:

  – that is acceptance testing

What does Unit Testing Accomplish ?

- *Does the **code** do what was expected?*

  – i.e. is the code fulfilling the intent of the developer?

- *Does the **code** do what was expected all the time*?

  – exceptions get thrown, disks get full, network lines drop, buffers overflow - is the code still performing as expected?

- *Can the **code** be depended upon?*

  – Need to know for certain both its strengths and its limitations.

How is Unit Testing carried out?

- **Step 1**: Decide how to test the method in question before writing the code itself.

- **Step 2**: Write the test code itself, either before or concurrently with the implementation code.

- **Step 3**: Run the test itself, and probably all the other tests in that part of the system.

Key Feature of executing tests: need to be able to determine at a glance whether all tests are succeeding/failing.

Topic List

- – What is Test Driven Development?

- – What is Unit Testing?

- – The JUnit Framework.

JUnit

- JUnit is a unit testing framework for the Java programming language.

- JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit.

JUnit Features

- Open source framework used for writing & running tests.

- Provides *Annotations* to identify the test methods.

- Provides *Assertions* for testing expected results.

- Provides Test runners for running tests.

- JUnit tests can be run automatically and they check their own results and provide immediate feedback. There's no need to manually comb through a report of test results.

- Junit shows test progress in a bar that is green if test is going fine and it turns red when a test fails.

"Good programmers write code, great programmers write tests"

"Never, in the field of programming, have so many owed so much to so few"

- Martin Fowler on the developers behind JUnit