Test Driven Development

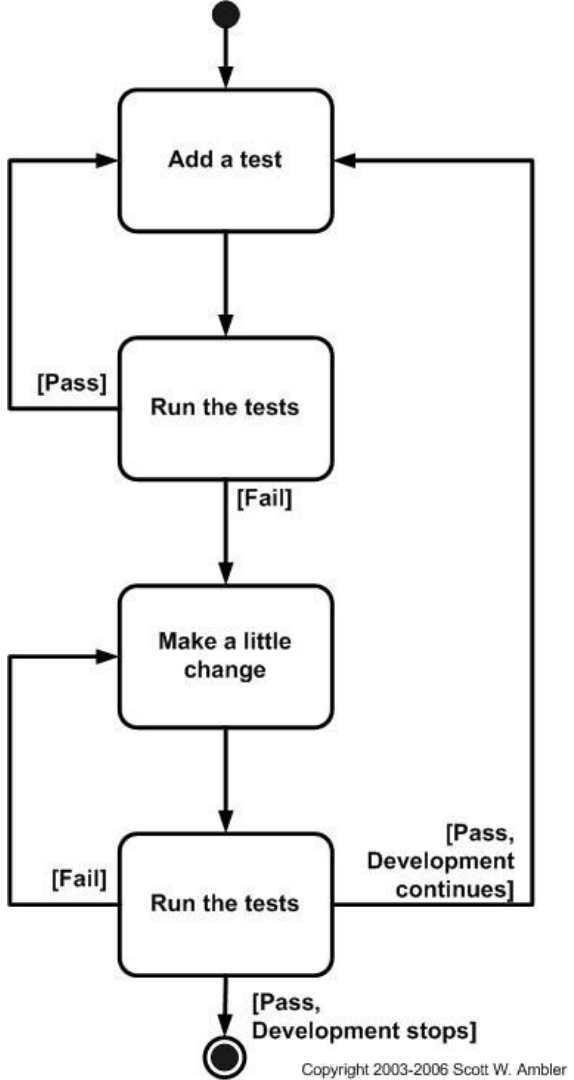
Unit testing in python

Test Driven Development

- There are many ways to run a software project. Agile methodology is one of them
- Is a part of Agile methodology to drive an efficient coding/development cycle

TDD Cycle

- Write the tests cases first
- Run them
- They will fail the first time since there is no code behind them
- Write code
- Run them
- Make changes - Refactor
- Run them
- Do it till all test cases pass



Use a unit testing framework

- Pyunit is one such and comes by default with Python
- Code walkthrough on how to use it

Available frequently used functions

- assert: base assert allowing you to write your own assertions
- assertEqual(a, b): check a and b are equal
- assertNotEqual(a, b): check a and b are not equal
- assertIn(a, b): check that a is in the item b
- assertNotIn(a, b): check that a is not in the item b
- assertFalse(a): check that the value of a is False
- assertTrue(a): check the value of a is True
- assertIsInstance(a, TYPE): check that a is of type "TYPE"
- assertRaises(ERROR, a, args): check that when a is called with args that it raises ERROR

Bingo Rhyme

- Code walkthrough
- Needs basics of OOP
- Next try the calculator

Advantages

- Saves time during the testing cycle, since most of the testing happens during development
- Better design of the code - Test cases drive the need for refactoring and a better design of the code
- Good quality code
- Good test coverage (the breadth of test cases written)

Disadvantages

- Development cycle is slower because of the test cases written by the developer alone
- This can be avoided by having another developer/Manager write the test cases and the developer write the