

Michael Eaton

Lab05

1. Write a test for a method, run it, fix any errors thrown. Rinse, repeat until method give desired action.
2. I can agree with this, provided that the developer writes a comprehensive suite of tests for his software. The method does almost nothing if inadequate testing is done.
3. One large advantage to Test Driven Development is you are absolutely sure that the method works for the tests you specified, and you have a large suite of tests to run and make sure it's behaving properly afterward. On the other hand, if you do things strictly by the book, you waste a lot of time making filler methods rather than implementing an algorithm you've already created straight away.