Test Driven Development

Using the IDE to make it easy

Words from Uncle Bob

“Why do developers fear to make continuous changes to their code? They are afraid they will break it! Why are they afraid they will break it? Because they don't have tests.”

Why don't they have tests

1. Tests were an optional aspect of the project
2. Testing is done by the QA team
3. Tests are not of value to the users (until it breaks)
4. The application is too coupled to test (Sandro???)
5. We ignore the tests anyway
6. Other dumb reasons ...

Is this really a professional approach to working?

Why TDD

1. Test creation is implicit, thus high code coverage
2. Functional code only focuses on known use cases (est 45% of functionality delivered is actually never executed)
3. Focuses system design on behaviour
4. Less debugging (failing tests tell you all you need to know)
5. Fast feedback
6. Faster development cycles
7. Refactor with confidence

What do you think?

Rules of the game

Not allowed to write any functional code until you have a failing test

Not allowed to write more of a unit test than is sufficient to fail (not compiling is failing)

You are not allowed to write more functional code than that which is needed to get the test passing

RED → GREEN → REFACTOR

What makes a good unit test?

1. Small ... think order of three lines
2. Concise … only test one behaviour
3. Fast … if it runs really fast you are likely to run them more often
4. Documentation … a well named test will self document the code

Test that ...

“Test That <classUnderTest> [test name]”

Some examples:

Test that Adder should add two numbers together to form the sum.

```
public void shouldAddTwoNumbersTogetherToFormTheSum()
```

Test that Adder should throw an exception when passed null numbers

```
public void shouldThrowExceptionWhenPassedNullInput()
```

OK lets have a go ...

Form pairs ... or as best you can

Start your favourite IDE .. prizes for choosing Vi

I was going to write a cheat sheet to make you use that for reference .. it was at the bottom of the backlog ...

Game rules

1. Not allowed to write functional code without a test
2. Only allowed to write as much of a test as to fail
3. Only allowed write enough functional code to pass test
4. Not allowed to use the mouse (gotta learn the keystrokes)
5. Not allowed to physically write the public or private class API (only the implementation) in the functional code … got to drive from the tests

Story 1

As a …

I need to calculate the price of a stock portfolio

So that I can understand the value of my investment

Acceptance criteria:

For a given stock 'FOO.L' I can calculate the price of the portfolio

Note: portfolio price is number of stocks * price of stock

Story 2

As a …

I need to calculate the price of a stock option portfolio

So that I can understand the value of my investment

Acceptance criteria:

For a given stock 'BAR.L' I can calculate the price of the portfolio

Note: [very simple] option price is number of options * (current price – strike price)

Story 3

As a …

I need to calculate the price of a mixed portfolio

So that I can understand the value of my investment

Acceptance criteria:

For a given options of 'BAR.L' and stocks of 'FOO.L' I can calculate the price of the portfolio