

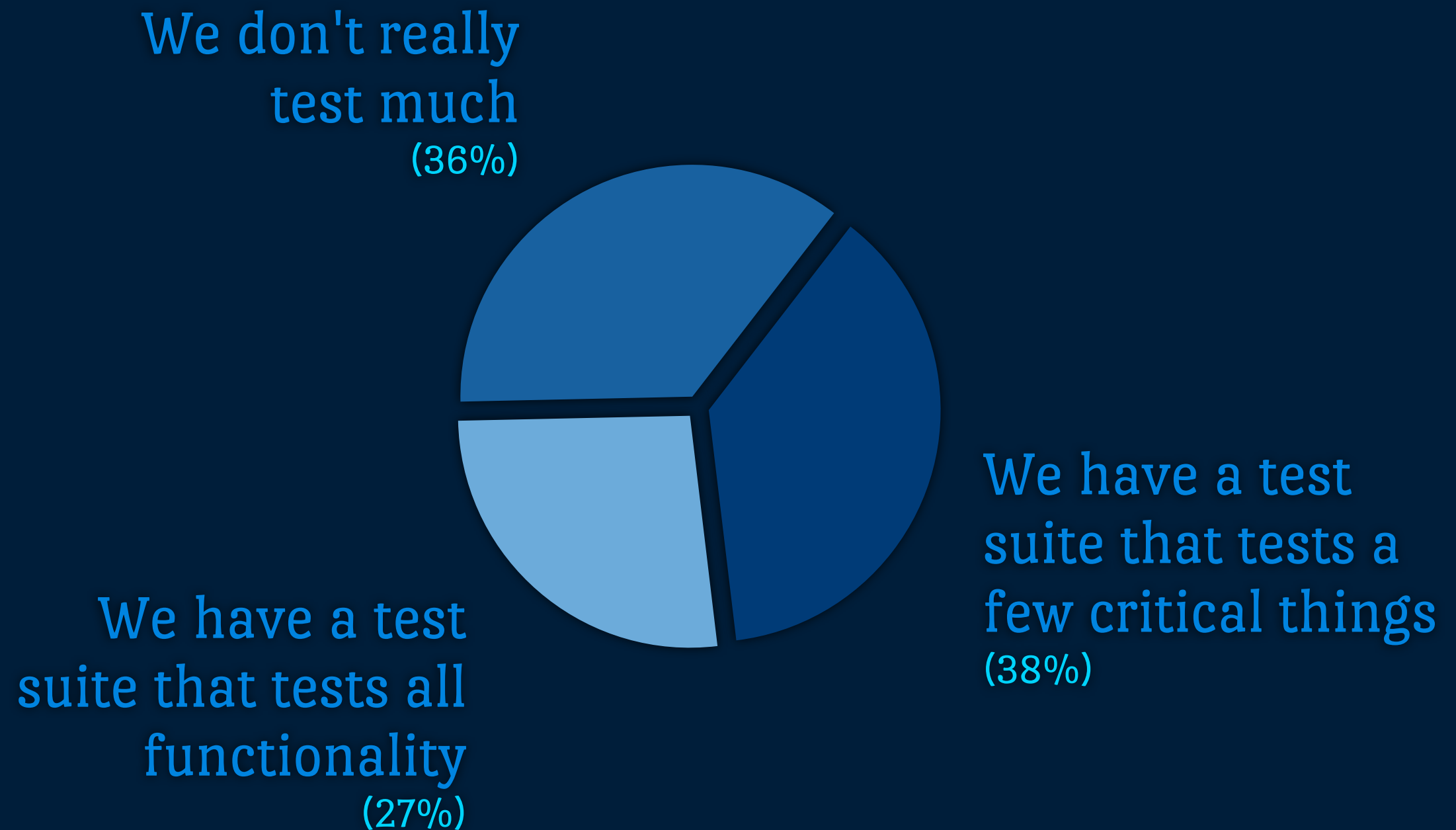
behaviour driven
(development/design)

in

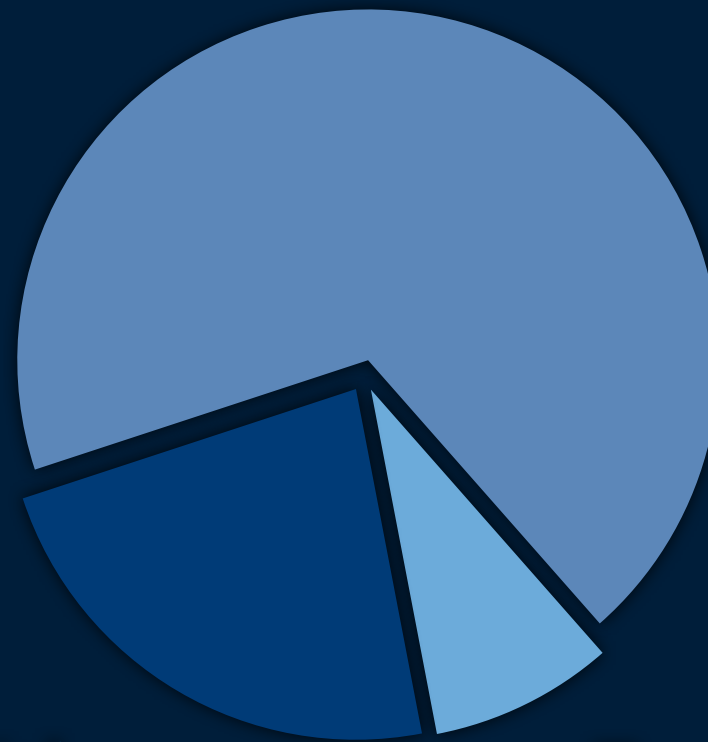
.NET

Hughes

James



We'd like to do more
testing but it's too
much overhead
(69%)



We are happy with
the amount of
testing we do
(23%)

Tests? We don't
need no stinking
tests.
(8%)

“

We, as an industry
don't feel we write
enough tests

”



too much effort, too much
friction, not enough time,
fragile tests, too little
incentive, testing the
wrong things, too much
focus on the machinery ...

behaviour
driven
(development
| design)

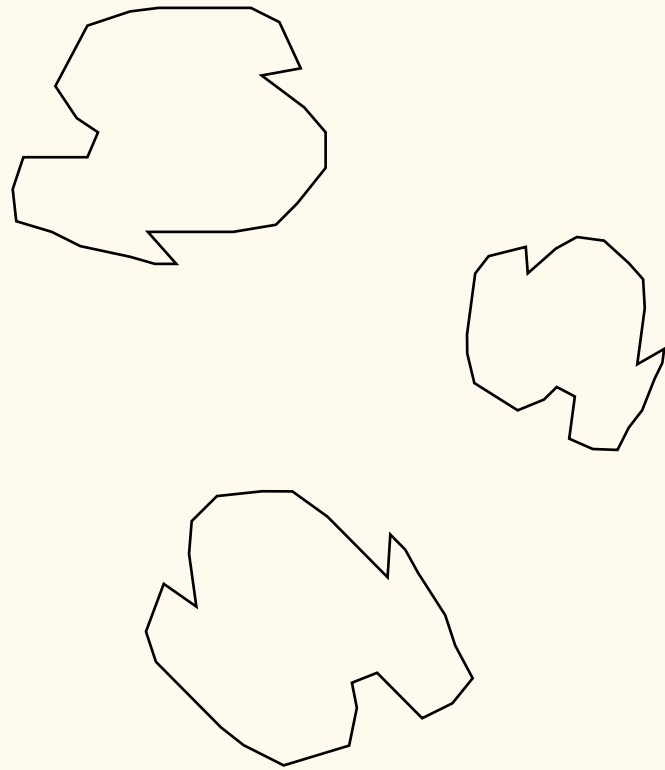
bring the customer with
you, build testing into
every stage, reduce
system testing cycle
time, stable tests, test the
right things, move the
focus to the behaviour ...



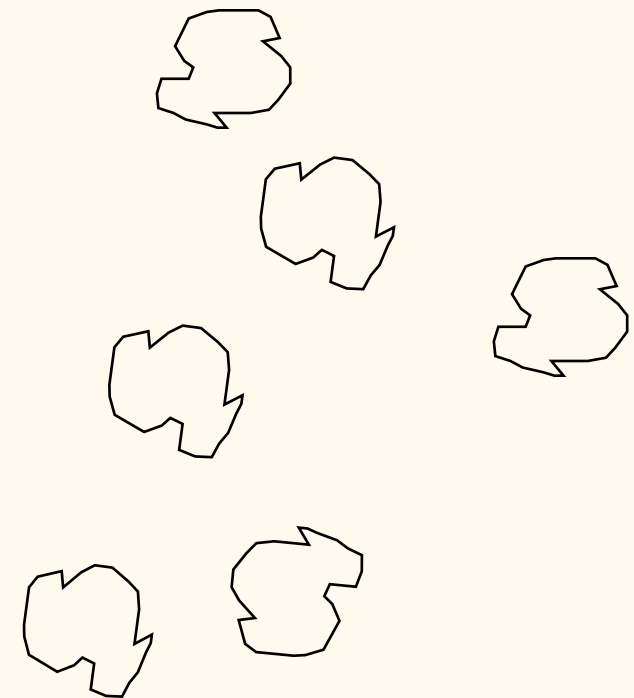
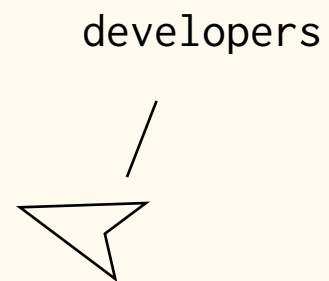
concepts



PRESS <START> TO PLAY



coarse grained
(acceptance tests)

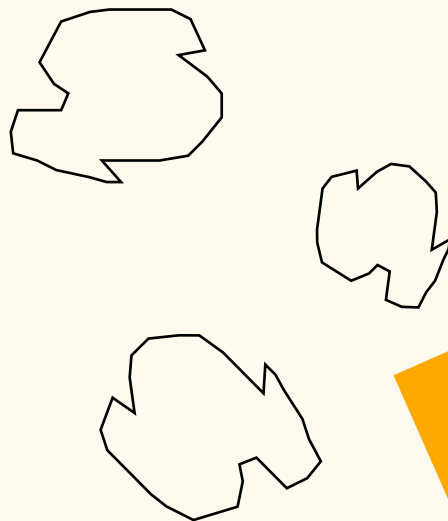


fine grained
(unit tests)

Given I complete the login form

When I submit the form

Then I should be logged in



coarse grained

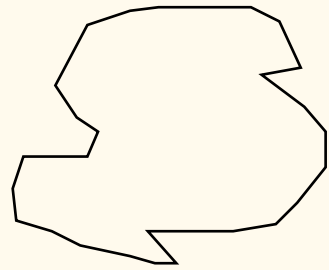
```
class describe_Account : account_spec
{
    void before() { account = new Account(); }

    void describe_withdrawing_cash()
    {
        context["when account is in credit"] = () =>
        {
            before = () => account.Balance = 500;

            it["the Account dispenses cash"] = () =>
                account.CanWithdraw(60).should_be_true();
        };
    }
}
```

fine grained

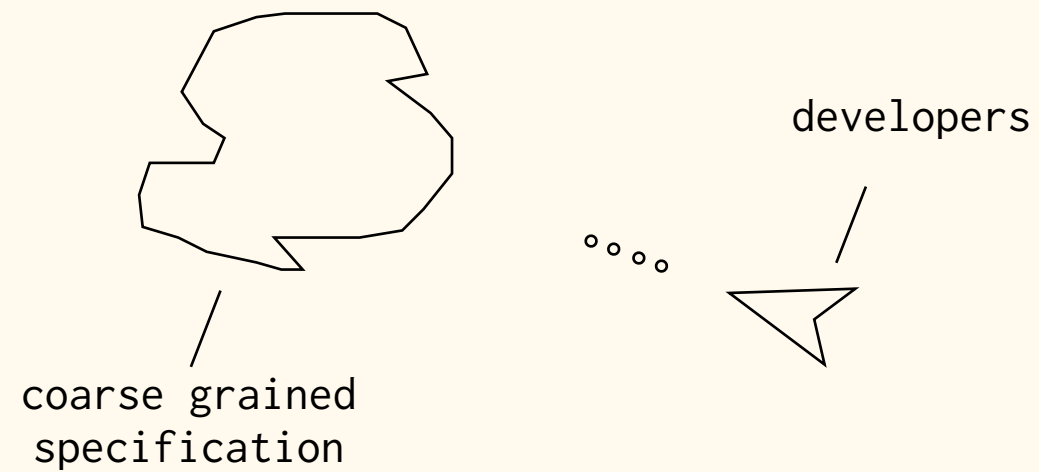




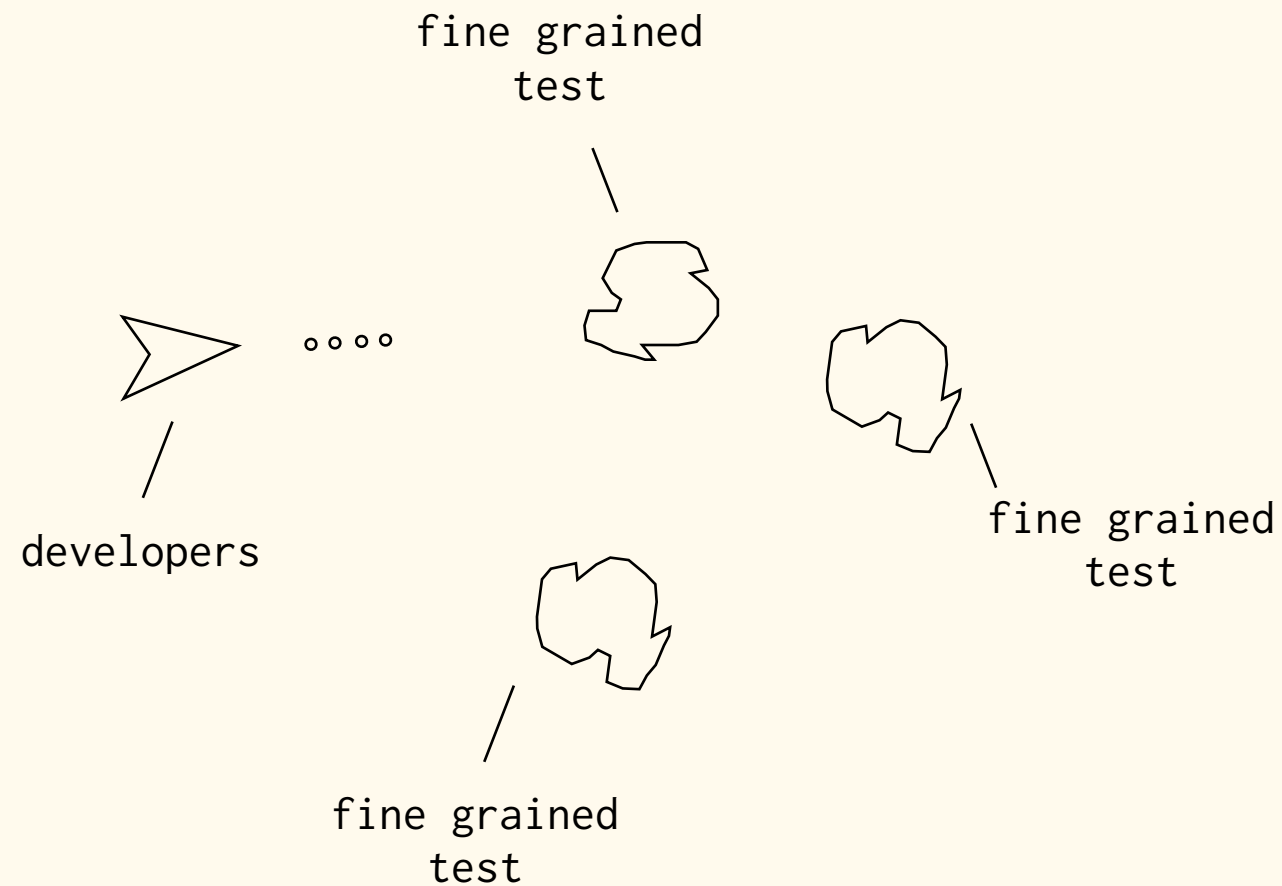
bdd (development) cycle
aka
RED, red, green, refactor,
GREEN, REFACTOR

developers

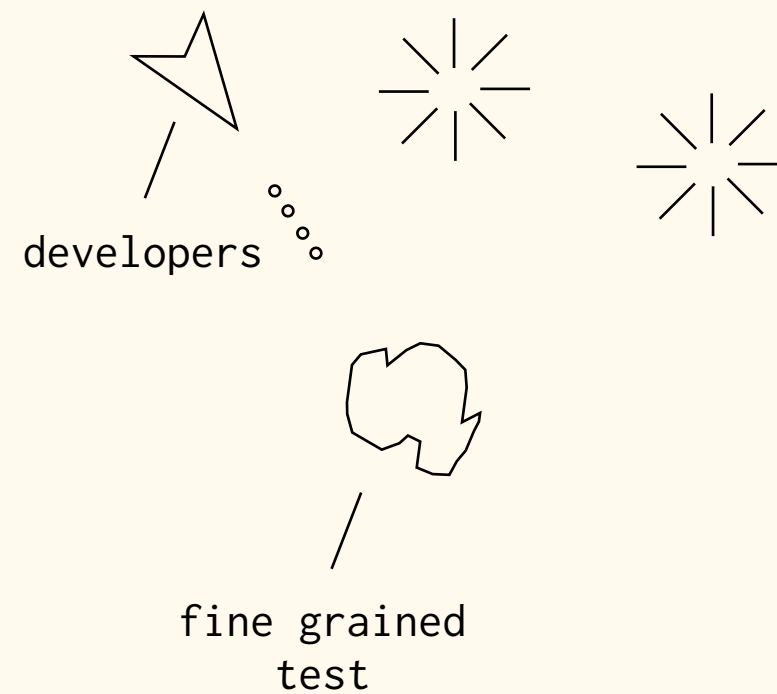




bdd (development) cycle
aka
RED, red, green, refactor,
GREEN, REFACTOR



bdd (development) cycle
aka
RED, red, green, refactor,
GREEN, REFACTOR

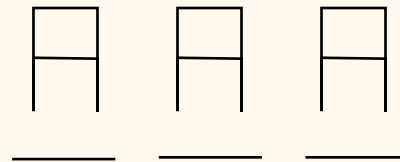


bdd (development) cycle
aka
RED, red, green, refactor,
GREEN, REFACTOR



SPECIFICATION OVER

NEW HIGH SCORE



PRESS <START> TO PLAY AGAIN

bdd (development) cycle

aka

RED, red, green, refactor,
GREEN, REFACTOR

StorEvil

Cuke4Nuke

Cucumber

StoryQ

XUnit

MSpec

NUnit

SpecFlow

coarse grained

NSpec

fine grained

.NET

The Calculate-o-tron

- A study into the art of BDD -

Feature: The Add-o-tron module of the Calculate-o-tron
So that I can win at maths
As someone who is too lazy to think for themselves
I want to be able to do maths on two numbers

Scenario: Calculating the sum of two numbers
Given I have navigated to the Add-o-tron
And I have entered 60 and 50
When I activate the add-o-tron
Then the result should be 110

coarse grained

```
[Given("I have navigated to the (.*)")]
public void GivenIHaveNavigatedTo(string function)
{
    browser.Navigate("http://localhost/DevWeek.Demo.Web");
    browser.Find("a", FindBy.PartialText, function).Click();
}
```

```
[Given("I have entered (.*) and the (.*)")]
public void GivenIHaveEnteredValues(string num1, string num2)
{
    browser.Find("number1").Value = num1;
    browser.Find("number2").Value = num2;
}
```

```
[When("I activate the (.*)")]  
public void WhenIPressActivateTheMachine(string function)  
{  
    browser.FindAll("form").First().SubmitForm();  
}
```

```
[Then("the result should be (.*)")]  
public void ThenTheResultShouldBe(string expectedResult)  
{  
    string actualResult = browser.Find("result").Value;  
    Assert.That(actualResult, Is.EqualTo(expectedResult));  
}
```

```
class describe_Add0TronService : nspec
{
    Add0TronService service;
    int actual;

    void before_each()
    {
        service = new Add0TronService();
        actual = 0;
    }

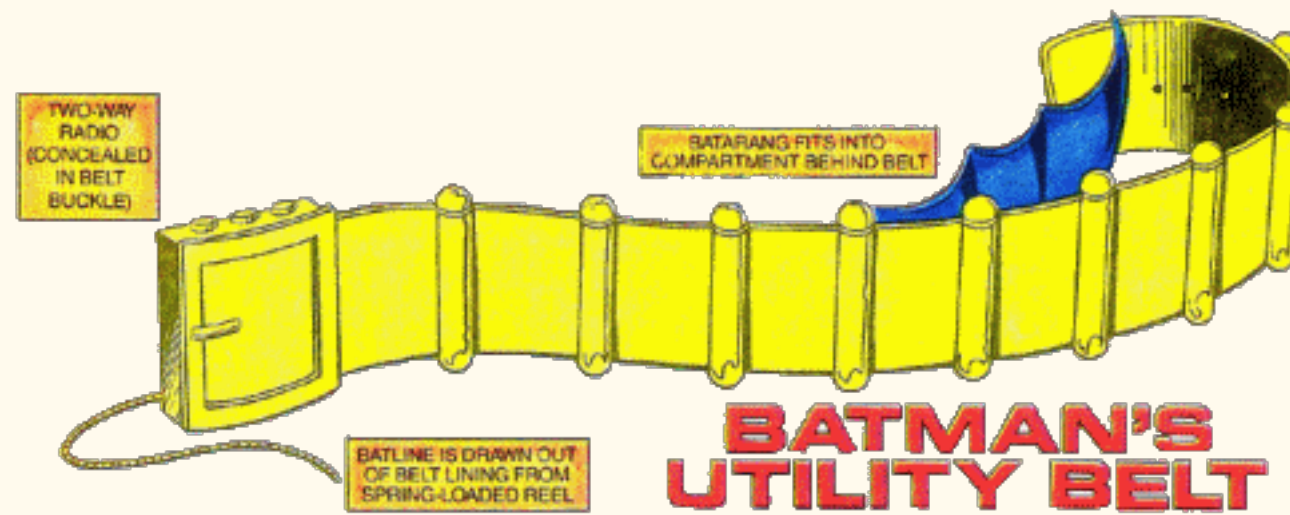
    void given_i_pass_in_2_valid_numbers()
    {
        act = () => actual = service.Calculate(45, 30);

        it["should add the numbers correctly"] = () =>
            actual.should_be(75);
    }
}
```

fine grained


```
describe AddOTronService
  when I pass in 2 valid numbers
    it should add the numbers correctly
```

1 Examples, 0 Failed, 0 Pending



Test Runners



testdriven.net

manual



resharper

manual



mighty moose
automatic



specwatchr
automatic



ncrunch
automatic

And now for something
completely different...

A yellow square logo with the letters 'JS' in a dark blue, bold, sans-serif font. The logo is centered on a dark blue background.

JS



```
test("QUnit", function() {  
  ok( true, "makes testing JavaScript possible" );  
  var value = "code";  
  equal( value, "code", "We expect your code to be better" );  
});
```

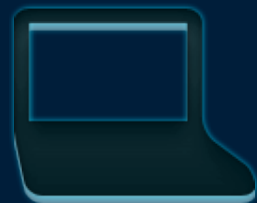


```
describe("Jasmine", function() {  
  it("makes testing JavaScript awesome!", function() {  
    expect(yourCode).toBeLotsBetter();  
  });  
});
```

< /Presentation >



bit.ly/bddindotnet-slides



bit.ly/bddindotnet-code

[@kouphax](#) / james@yobriefca.se