Test Driving .NET

Keith Burnell

Senior Software Engineer
Falafel Software, Inc
@keburnell · DotNetDevDude.com

Little about me

- Microsoft MVP: ASP.NET/IIS
- Senior Software Engineer at Falafel Software, Inc.
- Been developing software for > 15 years
- Primary focus on the Web stack
- Speaker
- Author

Pop quiz

Ever have to deal with **spaghetti code?**

Ever inherit **legacy code?**

Ever **scared** to change code?

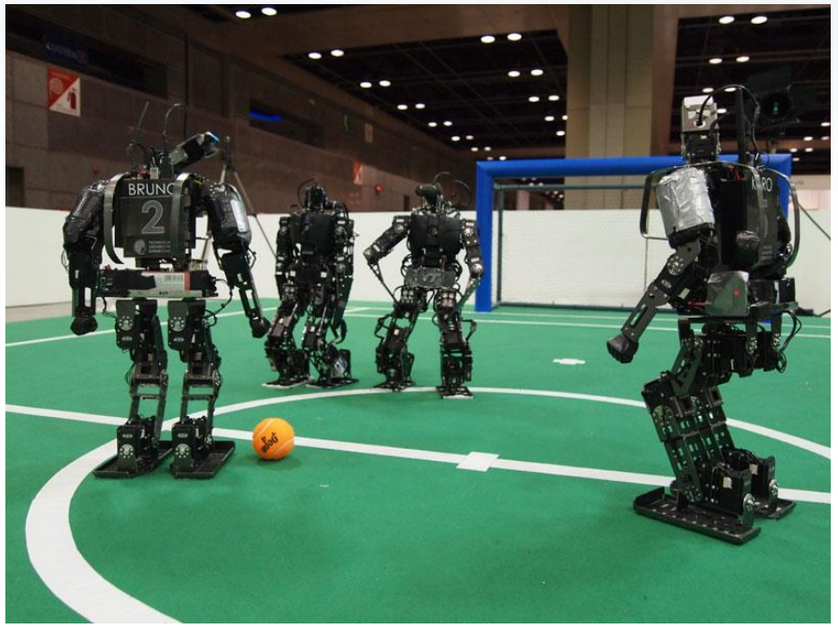
If you answered **YES** to any of those questions

Then…

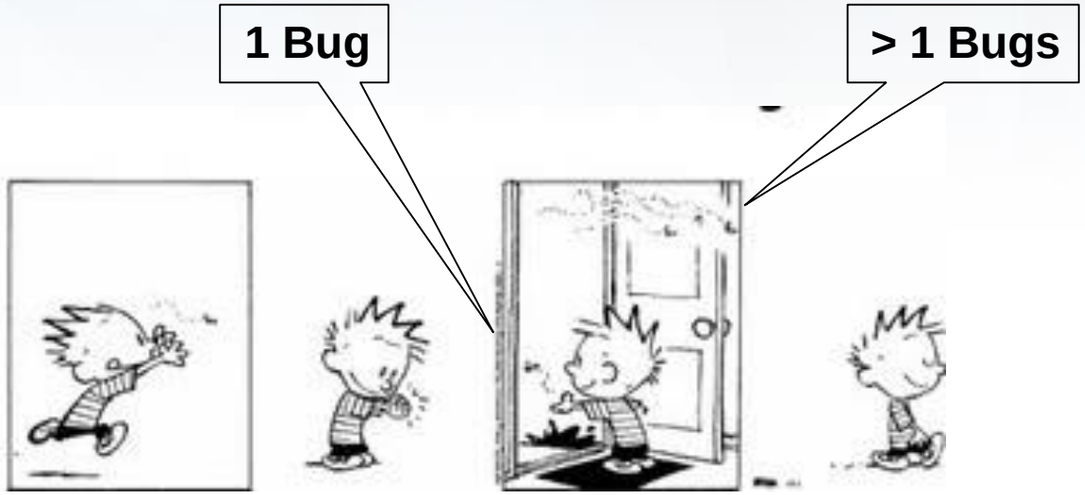

You need
some tests yo!
Dave Bouwman :: DTSAgile.com

Types of testing that **are** not keys to test-driven development

Integration Testing



Regression Testing

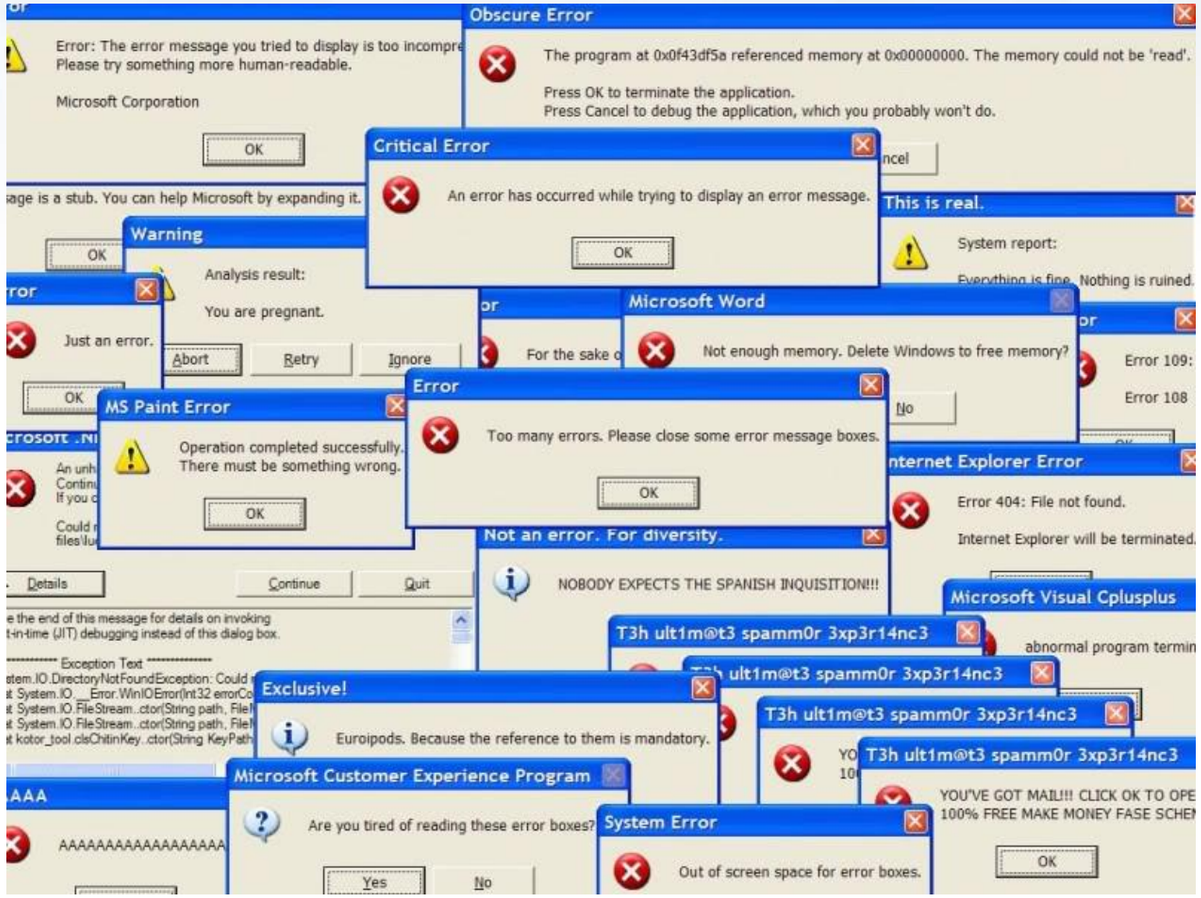


User Acceptance Testing (UAT)

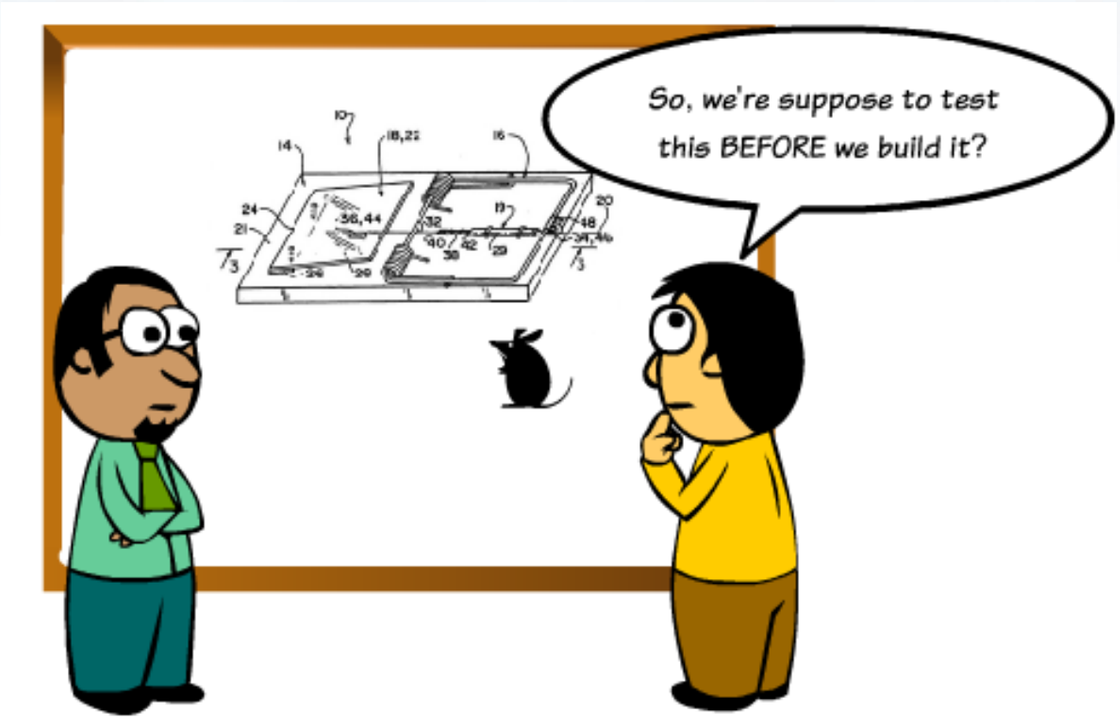
Performance Testing

Load Testing

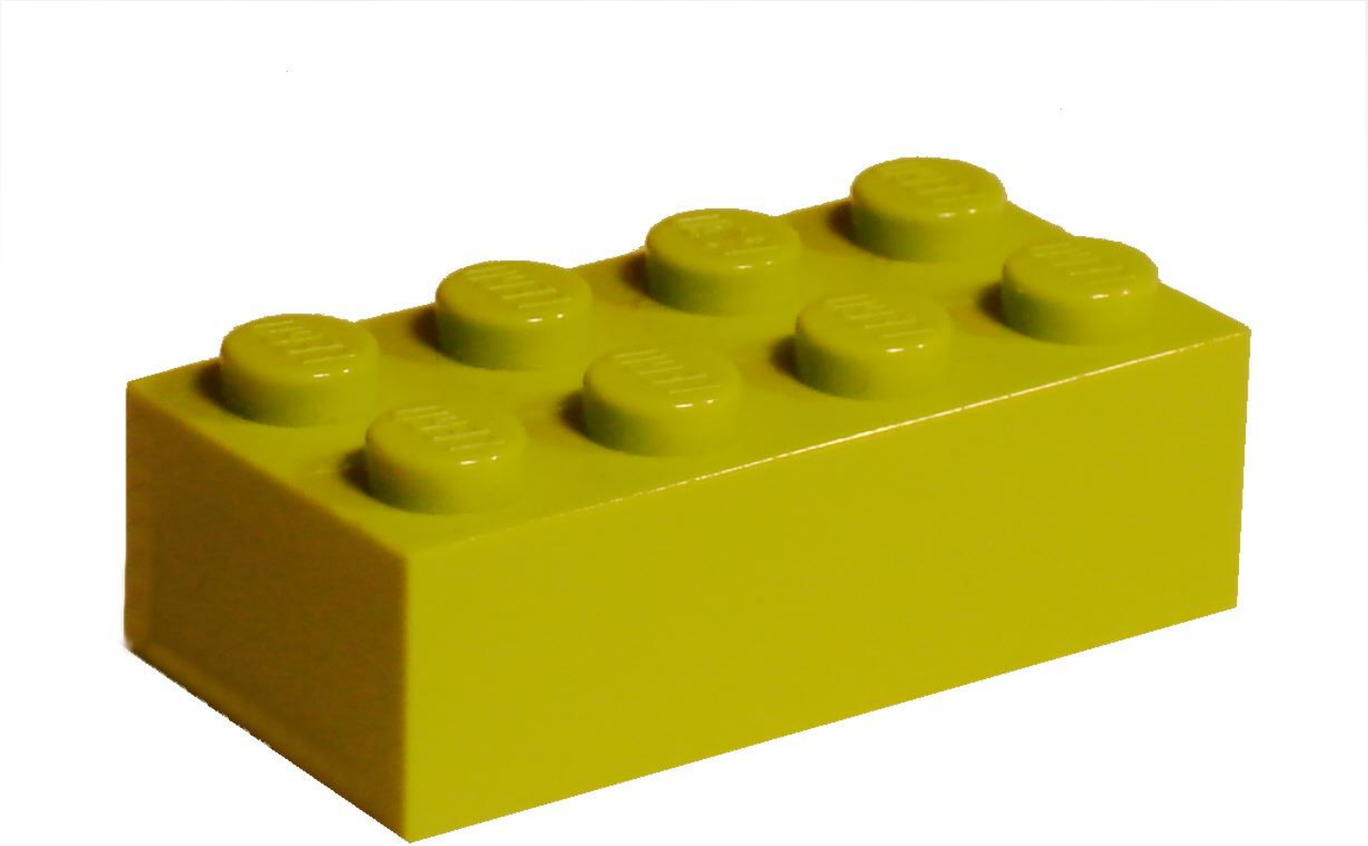
Stress Testing



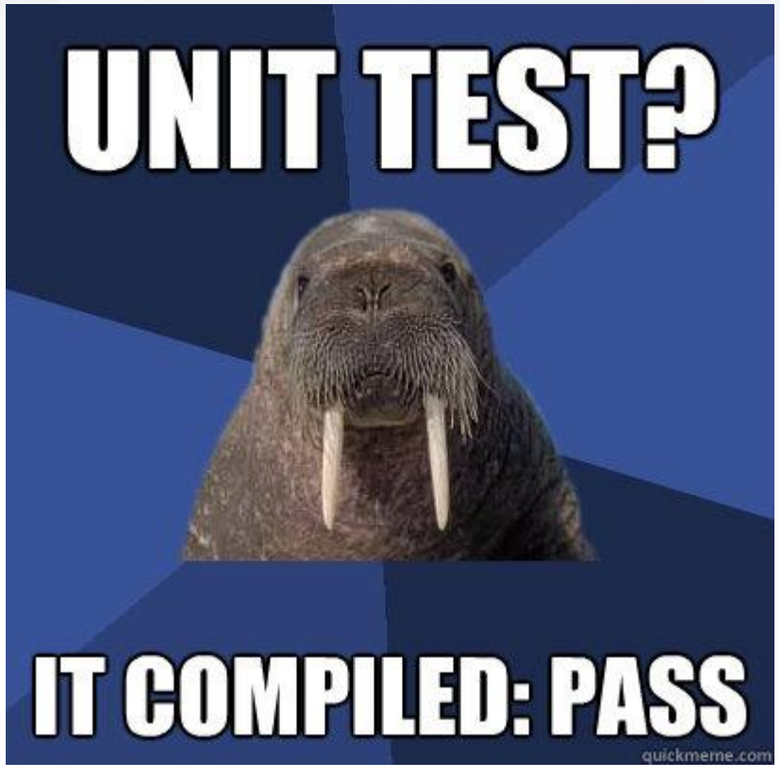
What is Test-driven Development?



Unit Testing: Key to TDD



What makes a good Unit Test?



Automated and Repeatable

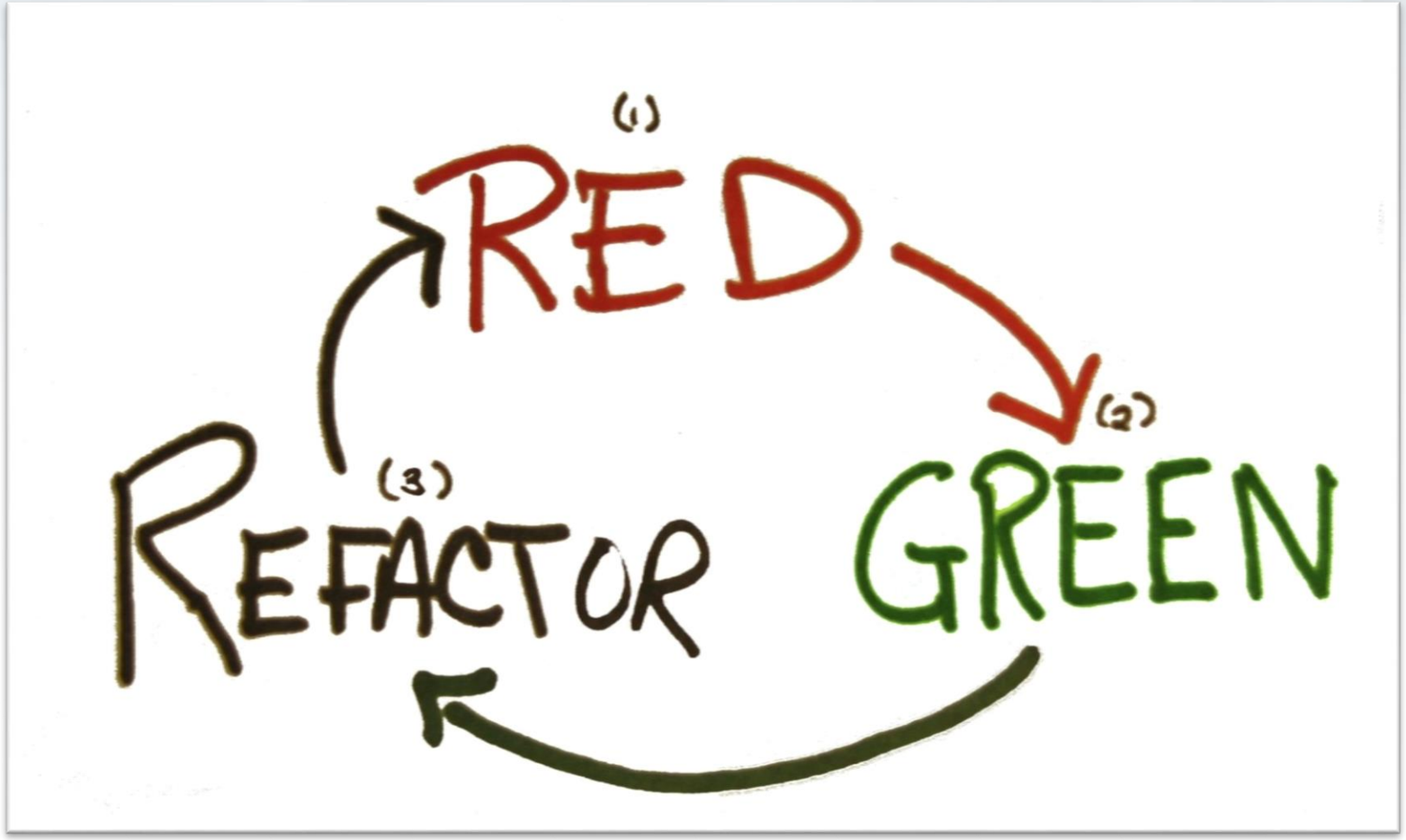
Easy to implement

Run on Demand at the Push of a Button

Fast

Isolated

Important Concepts & Strategies


(1)
RED
(2)
GREEN
(3)
REFACTOR

Arrange – Act - Assert

SOLID Principles of Software Development

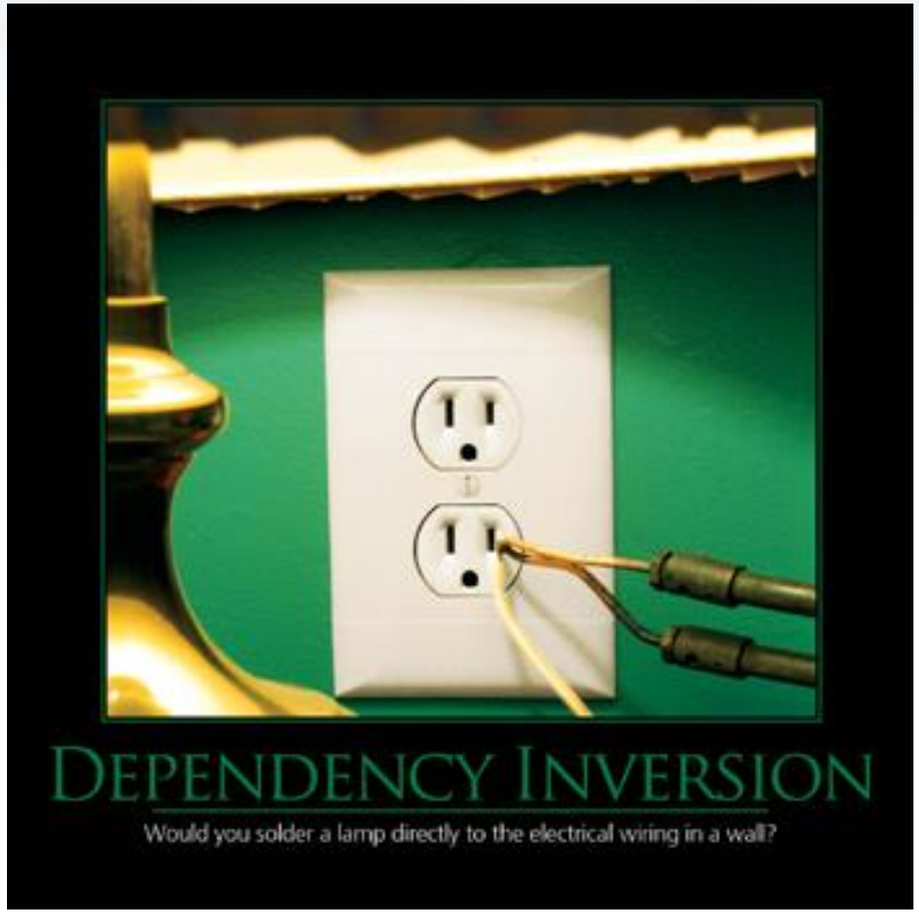
Single Responsibility Principle

Open Closed Principle

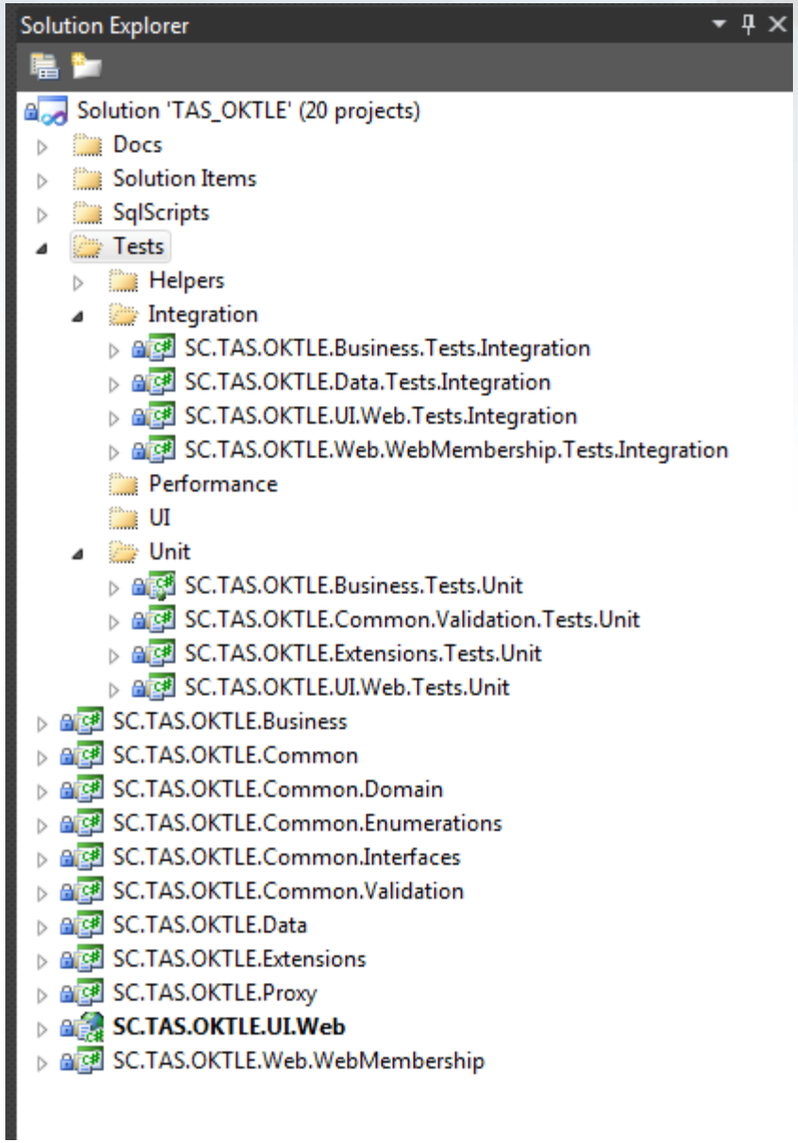
Liskov Substitution Principle

Interface Segregation Principle

Dependency Inversion Principle



Example Solution Layout



Naming Your Tests

How I Name My Tests

- [*MethodUnderTest*]_[*ExpectedResult*]_[*Conditions*]
- `GetCustomers_ShouldReturn_ListOf_Customers()`
- `CalculateRate_ShouldReturn_25_When_Interest_Is_50_And_X_Is_5()`

Resharper Live Template for Naming Tests

- My unit testing live template is available on my blog:
  - http://www.dotnetdevdude.com/downloads/code/ResharperUnitTestTemplates.zip

```
//tt -> tab
[TestMethod]
public void MethodBeingTested_ExpectedResult() {
    //Arrange

    //Act
    //Assert
}

//tw -> tab
[TestMethod]
public void MethodBeingTested_ExpectedResult_WhenWhat() {
    //Arrange

    //Act
    //Assert
}
```

Let’s Write Our First Unit Test - Together!

Dependency Injection (DI)

Inversion of Control (IoC)

What does implementing DI/IoC get me?

- Separation of Concerns

- Mockability (we will get to this)

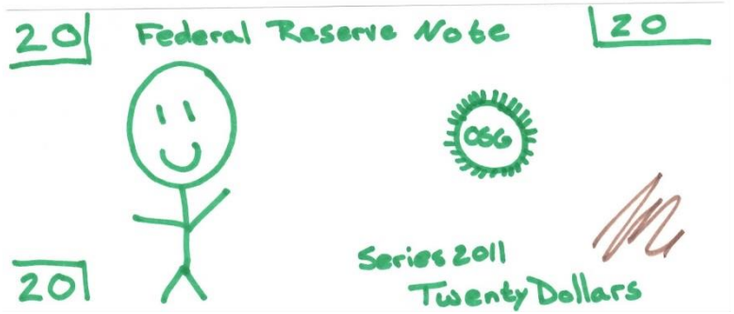
Nicole Weston

Let’s Do Some DI/IoC - Together!

Ruh roh!
Test is broken!

Fakes/Stubs


20
FEDERAL RESERVE NOTE
20
IE 54380694 C
E5
THE UNITED STATES OF AMERICA
IE 54380694 C
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
SERIES 2006
20
JACKSON
TWENTY DOLLARS


20
Federal Reserve Note
20
056
Series 2011
Twenty Dollars
20

Mocks

DI/IoC Tooling

Let’s Implement DI/IoC with Tooling - Together!

Other Resources

- Test Driven Development: By Example (Kent Beck)
- The Art of Unit Testing (Roy Osherove)
- Roy Osherove's Blog (http://www.osherove.com)

Tools I Used and/or Talked About

- NuGet (http://www.nuget.org)
- ReSharper (http://www.jetbrains.com/resharper/)
- Should Assertion Library (http://should.codeplex.com)
- NCrunch (http://www.ncrunch.net)
- RhinoMocks (http://hibernatingrhinos.com/open-source/rhino-mocks
- StructureMap (http://www.structuremap.net)

Thank You!

- Slides and Code:
  - github.com/KBurnell/TestDriving.NET
- Me:
  - Twitter: @KeBurnell
  - Blog: DotNetDevDude.com
  - E-Mail: Keith@DotNetDevDude.com