SAND2011-3900C

# Test Driven Development In .NET

Scott Griffin & Ramona Gallegos
Sandia National Laboratories

NLIT Summit 2011
June 17, 2011

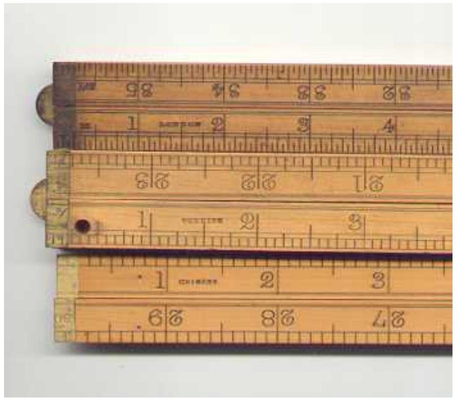
## Agenda

- What is TDD
- TDD is a design methodology, not a way of getting developers to write unit tests
- Why you should use TDD on your project
- Examples in .NET
  - Mocks

## How do you develop?

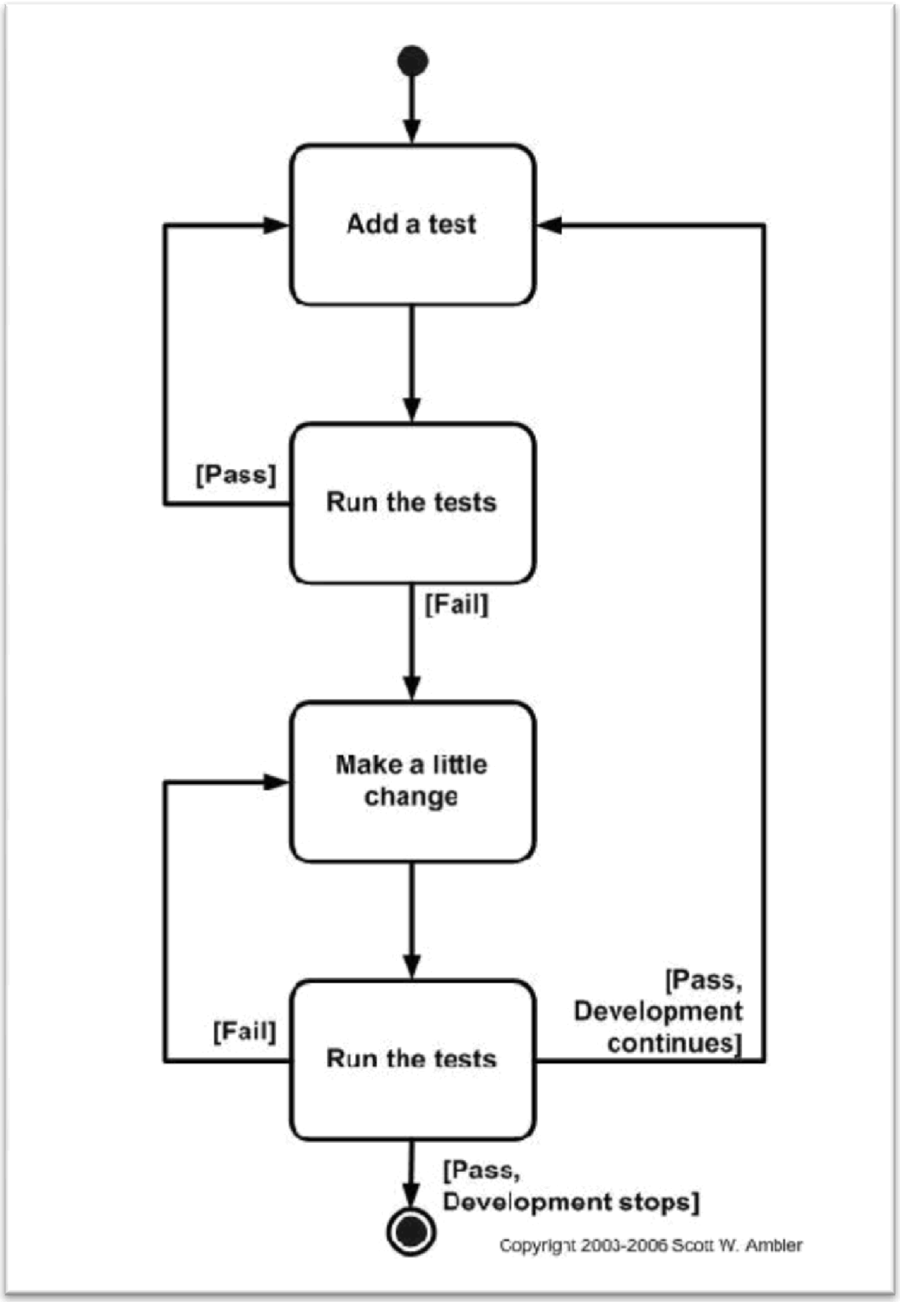
- How do you write new code?
  - Develop
  - Test
  - Debug
- Would you be willing to switch to:
  - Test
  - Develop
  - Debug

# Three Rules of TDD



- You are not allowed to write any production code unless it is to make a failing unit test pass.
- You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
- You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

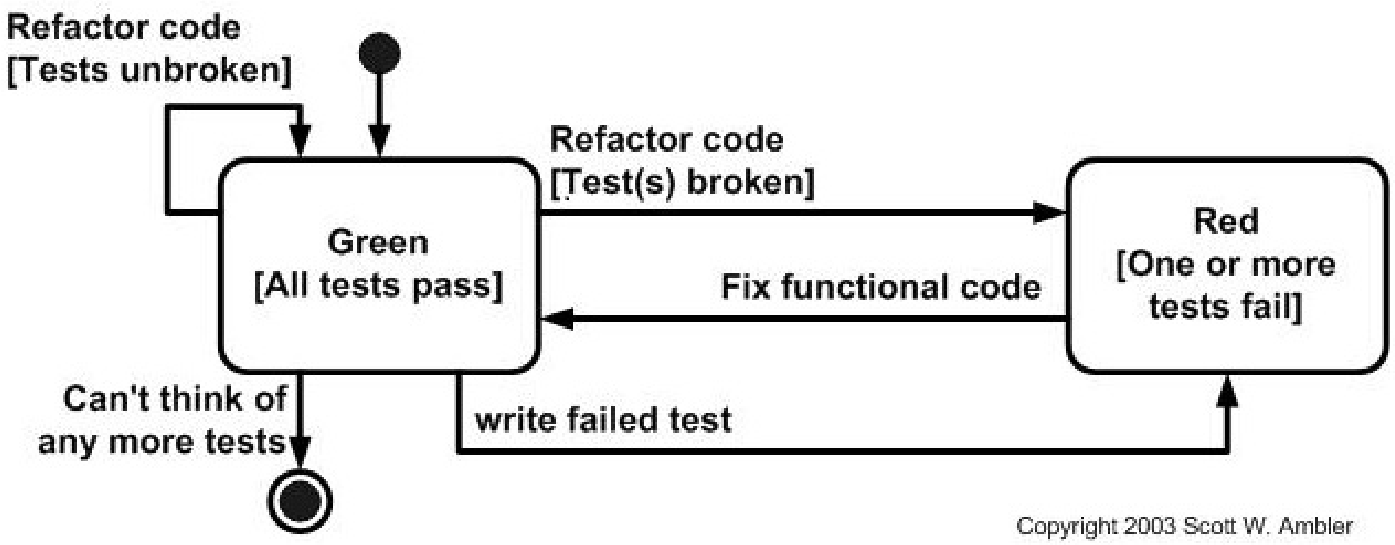
- **TDD?**
- Is this a reasonable representation of TDD?



Add a test
Run the tests
[Pass]
[Fail]
Make a little change
Run the tests
[Fail]
[Pass, Development continues]
[Pass, Development stops]
Copyright 2003-2006 Scott W. Ambler

# TFD versus TDD

- Previous slide is actually only Test First Design.
- What's the difference?
  - Refactoring



Refactor code [Tests unbroken]
Refactor code [Test(s) broken]
Green [All tests pass]
Fix functional code
Red [One or more tests fail]
Can't think of any more tests
write failed test
Copyright 2003 Scott W. Ambler

## Why is Refactoring important?

- You re-examine the design every time a feature is added.
  - Is my design correct for this feature?
  - If yes, write a test.
  - If no, refactor.
- You design organically, with the running code providing feedback between decisions.

Sandia National Laboratories

## TDD isn’t about testing

- Testable code is more modular.
- Testable code is loosely coupled.
- Your designs must consist of highly cohesive, loosely coupled components (e.g. your design is highly normalized) to make testing easier (this also makes evolution and maintenance of your system easier too).
- TDD makes your code better.

## But it takes more time!

- “If it's worth building, it's worth testing. If it's not worth testing, why are you wasting your time working on it?” – Scott W. Ambler

- “[W]hat would happen if you walked in a room full of people [using TDD]. Pick any random person at any random time. A minute ago, all their code worked. Let me repeat that: A minute ago all their code worked!” – Uncle Bob

- Our current project has over 2,000 unit tests.
- I am comfortable making any change. I know I can prove the code is working again when I'm done.
- I am fearless when changing other developer's code, because I know they wrote tests.

## What’s the best documentation?

- When importing a new library, do you look at documentation or examples?
- Unit tests are examples for future developers.

## Code!!

## References

- http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd
- http://www.agiledata.org/essays/tdd.html#WhatIsTDD

## Resources

- http://www.asp.net/
- http://stackoverflow.com/
- The Art of Unit Testing with Examples in .NET, Roy Osherove
- Test-Driven Development by Example, Kent Beck
- Rhino Mocks
- Fake It Easy

# Contact Information

Scott Griffin
Sandia National Laboratories
PO Box 5800, MS-0931
Albuquerque, NM 87185-0931
(505) 263-4948
scott.griffin@sandia.gov
@sgriffinusa

Ramona Gallegos
Sandia National Laboratories
PO Box 5800, MS-0931
Albuquerque, NM 87185-0931
(505) 284-8866
rkgalle@sandia.gov
@rkgallegos

Sandia National Laboratories