Unit testing - wounds of war

Codecamp Cluj - October 2011
Magdas Adrian
http://ullink.com/

Who am I

- .NET senior developer @ULLINK
- @amagdas on Twitter
- magdasadrian@gmail.com on Google+

Agenda

1. Unit tests - what, how and why
2. Types of tests
3. Tools of the trade, quick intro
4. Principles of good unit testing
5. Opinionated
6. Demo

Unit tests - what, how and why

- what is a unit test: “Unit of work testing”
- mantra: “Red, green, refactor”
- why do we need verifiable code
- how: SOLID principles, DIP and poor man's DI, patterns
- main benefits: feature design & playing ground, regression testing, quick feedback on refactoring

Types of tests or what is not a unit test

- unit
- integration(touches DataTime.Now, File, DB, Thread, Random)

Tools of the trade, quick intro

- unit testing frameworks: NUnit, MBUnit, MSTest etc.
- mocking: Rhino, NMock, TypeMock, Moq
- IOC containers
- continuous integration, code coverage

Principles of good unit testing

- repeatable
- naming conventions
- mocks vs stubs(state vs behavior)
- same rules as regular code
- easy to write
- FAST
- fake as little as necessary
- avoid fragile tests
- “Don’t hide a TUF(test unfriendly feature: DB, IO,) within a TUC(construct: static, constructor, sealed, non-virtual)

Opinionated - “Testing isn’t hard. Testing is easy in the presence of good design”

- test driven development vs. test after development
- test driven development vs. test driven design
- test driven development vs. behaviour driven development

Resources

- "Working Effectively With Legacy Code" - Michael Feathers http://www.objectmentor.com/resources/articles/WorkingEffectivelyWithLegacyCode.pdf
- "The art of unit testing" - Roy Osherove - http://artofunittesting.com/
- "Test driven development - by example" - Kent Beck - http://www.amazon.com/Test-Driven-Development-Kent-Beck/dp/0321146530
- "Agile Principles, Patterns and Practices in C#" - Robert C. Martin -