SOFTWARE CRAFTSMANSHIP

Clean Code:
Principles, Patterns, and Practices for Better Code

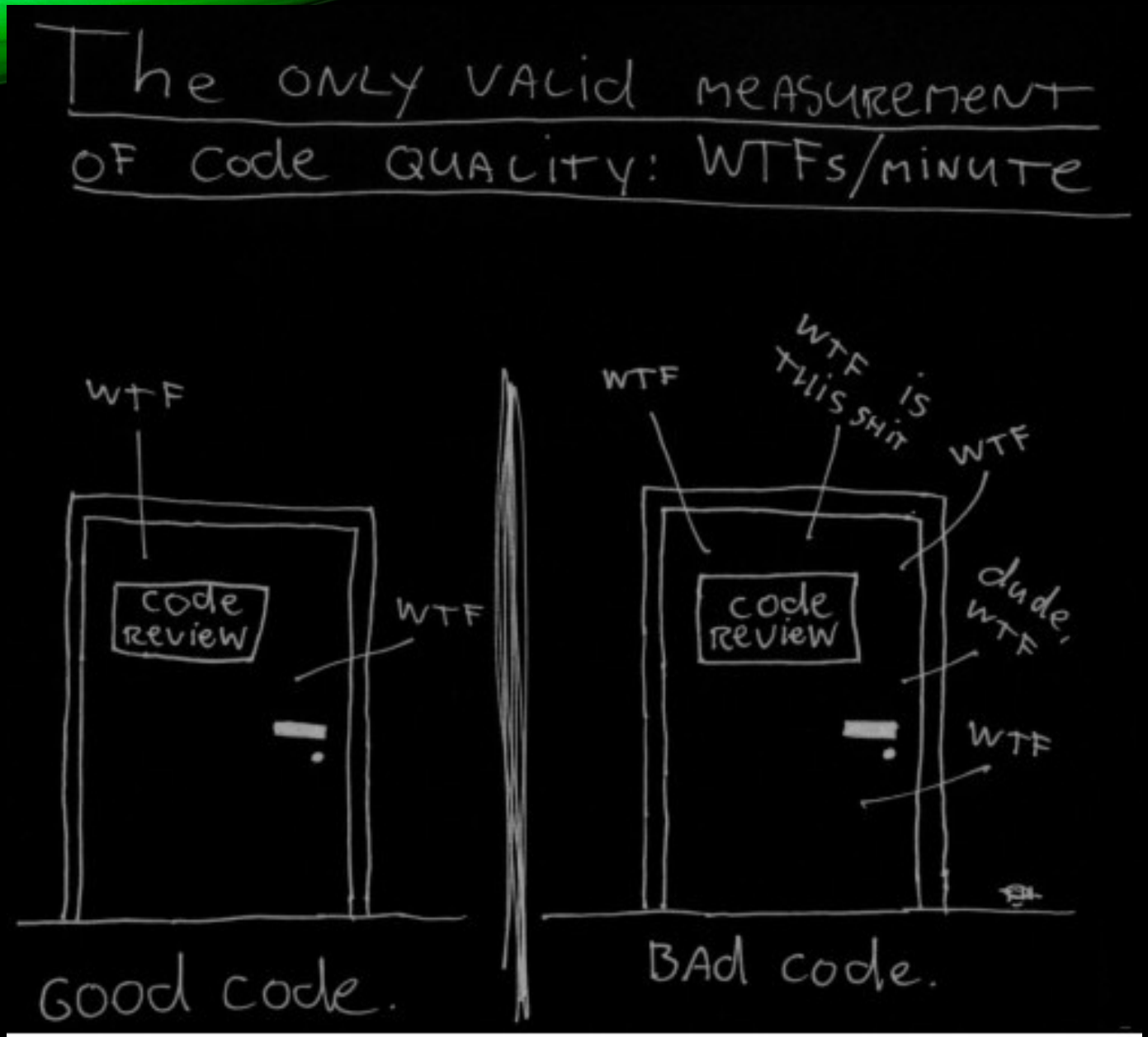
OVERVIEW

- What is Clean Code?
- Tools - Your Best Friend
- From Names to Classes
- The "Must Read"-Books
- Summary

© Theo Jungeblut 2014

WHAT IS CLEAN CODE?

©Jeremy Clark 2010



(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics

CLEAN CODE IS MAINTAINABLE

Source code must be:

- **readable & well structured**
- extensible
- testable

CODE MAINTAINABILITY *

Principles	Patterns	Containers
Why?	How?	What?
Extensibility	Clean Code	Tool reuse

* from: Mark Seemann's "Dependency Injection in .NET" presentation Bay.NET 05/2011

© Theo Jungeblut 2014

SOFTWARE ENGINEERING & SOFTWARE CRAFTSMANSHIP

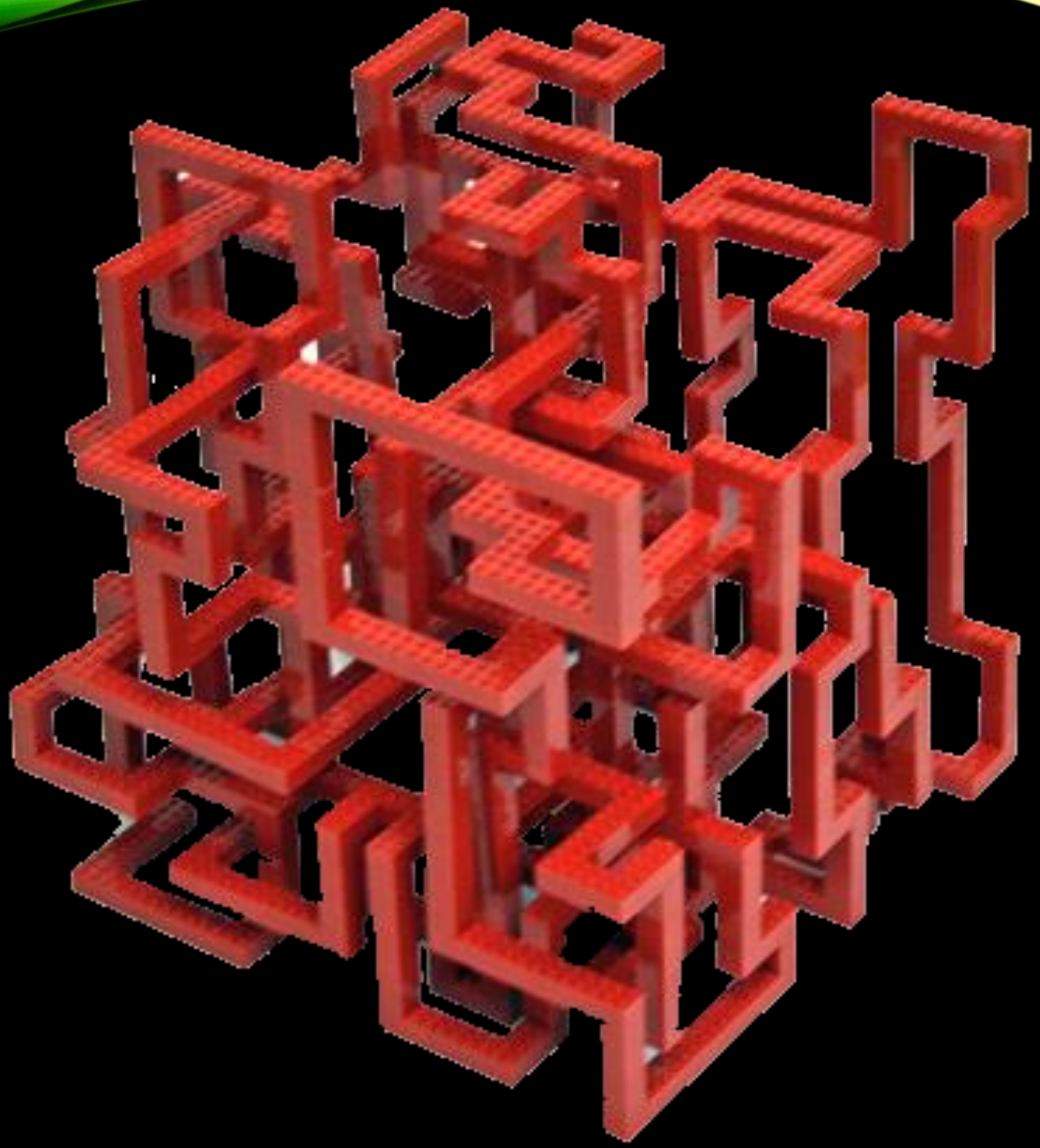
KEEP IT SIMPLE, STUPID (KISS)

“KEEP IT SIMPLE STUPID”

by Kelly Johnson

http://blogs.smarter.com/blogs/Lego%20Brick.jpg

© Theo Jungeblut 2014



Graphic by Nathan Sawaya courtesy of brickartist.com

© Theo Jungeblut 2014

SOURCE CODE CONVENTIONS

.NET TOOLS & THEIR IMPACT

Tool name	Positive Impact	Negative Impact
Resharper	compiling ++++, learning +++	VS responsiveness --
Code Analysis	code quality ++	compiling time -
StyleCop	code consistency +++	compiling time -
StyleCop plugin for Resharper	compiling time +++	VS responsiveness --
Ghost Doc	automated docs	potentially worse doc
Spell Checker	fewer spelling errors ++	performance --

© Theo Jungeblut 2014

RESHARPER

"The single most impacting development addition to Visual Studio"

Features:

- Code Analysis
- Quick Fixes
- Code Templates
- Code Generation
- Code Cleanup
- Many, many more…

http://www.jetbrains.com/resharper/

© Theo Jungeblut 2014

(STATIC) CODE ANALYSIS

Code Analysis:

- Correctness
- Library design
- Internationalization and localization
- Naming conventions
- Performance
- Security

http://msdn.microsoft.com/en-us/library/3z0aeatx.aspx

© Theo Jungeblut 2014

STYLE COP
WITH R# INTEGRATION

Code Consistency & Readability:

- Automated check of C# coding standard
- Enforceable at check-in with TFS check-in Policy

- Full Integration in Resharper with Style Cop plugin:
  - Code Analysis
  - Quick Fixes
  - Code Cleanup

© Theo Jungeblut 2014

Ghost Doc

- Save keystrokes and time
- Simplify documenting your code
- Benefit of the base class documentation
- Source code and text spellcheck *(Pro version)*
- API documentation creation *(Pro version)*

http://submain.com/products/ghostdoc.aspx

© Theo Jungeblut 2014

SPELL CHECKER

- Spelll chicking for literals and comments in VS

http://visualstudiogallery.msdn.microsoft.com/7c8341f1-ebac-40c8-92c2-476db8d523ce/

© Theo Jungeblut 2014

NAMES MATTER

- Meaningful Names
- Intention Revealing Names
- Use Pronounceable Names
- Use Searchable Names
- Avoid Encoding (Hungarian)
- Don't be cute
- Pick One Word per Concept
- Use Problem Domain Names

* Chapter extract: Robert C. Martin –" Clean Code", Parson Education, Inc. 2008

© Theo Jungeblut 2014

FUNCTIONS

- Small – Do One Thing
- One Level of Abstraction
- No or only few Arguments
- Have No Side Effects
- Prefer Exceptions to Returning Error Codes
- Don’t Repeat Yourself

* Chapter extract: Robert C. Martin –” Clean Code”, Parson Education, Inc. 2008

© Theo Jungeblut 2014

COMMENTS

- Comments do not Make Up for Bad Code
- Explain Yourself in Code
- Clarification
- Warning of Consequences
- ToDo Comments
- Javadocs in Public APIs

* Chapter extract: Robert C. Martin –" Clean Code", Parson Education, Inc. 2008

© Theo Jungeblut 2014

BAD COMMENTS

- Mumblings
- Redundant Comments
- Misleading Comments
- Journal Comments
- Noise Comments
- Don’t use a Comment When you Use a Method or a Variable
- Commented-Out Code

* Chapter extract: Robert C. Martin –” Clean Code”, Parson Education, Inc. 2008

© Theo Jungeblut 2014

THE PURPOSE OF FORMATTING

- Team Rules – Consistency is King
- Vertical Openness Between Concepts
- Vertical Distance
- Horizontal Alignment
- Indentation
- Write Journey Style Code

Resharper & StyleCop – "Code Cleanup"

* Kind of Chapter extract: Robert C. Martin –" Clean Code", Parson Education, Inc. 2008

© Theo Jungeblut 2014

CLASSES

- Class Organization
- Encapsulation
- Classes Should be Small
- The Single Responsibility Principle
- Cohesion
- Organize for Change
- Insolating from Change

* Kind of Chapter extract: Robert C. Martin –" Clean Code", Parson Education, Inc. 2008

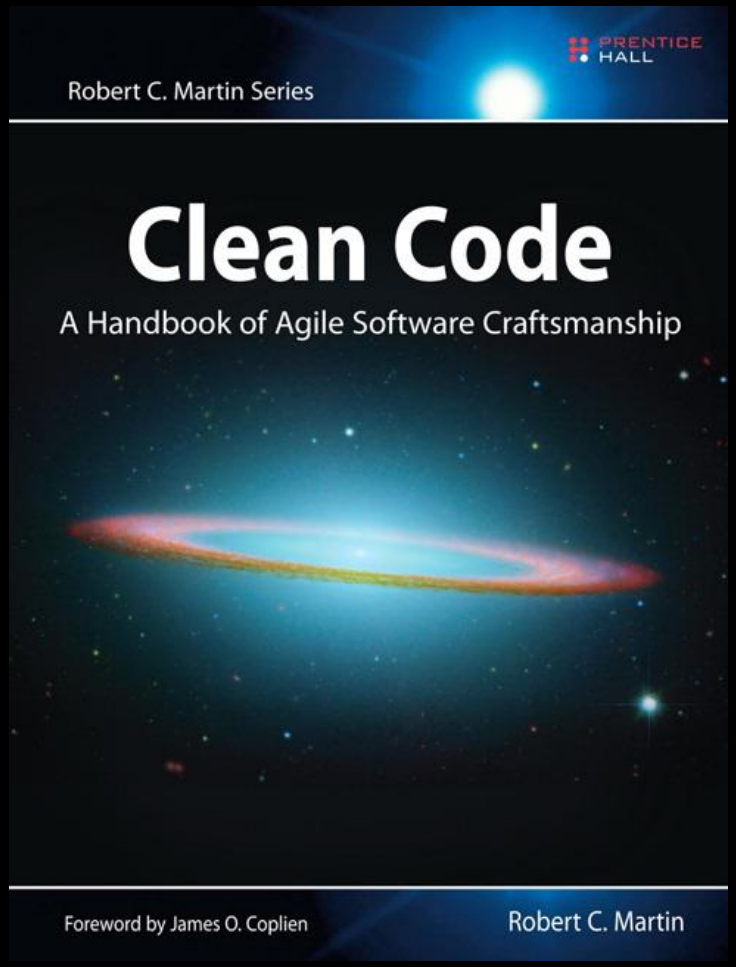
© Theo Jungeblut 2014

THE "MUST READ"-BOOK(S)

by Robert C Martin

A Handbook of Agile Software Craftsmanship

"Even bad code can function. But if code isn't clean, it can bring a development organization to its knees."



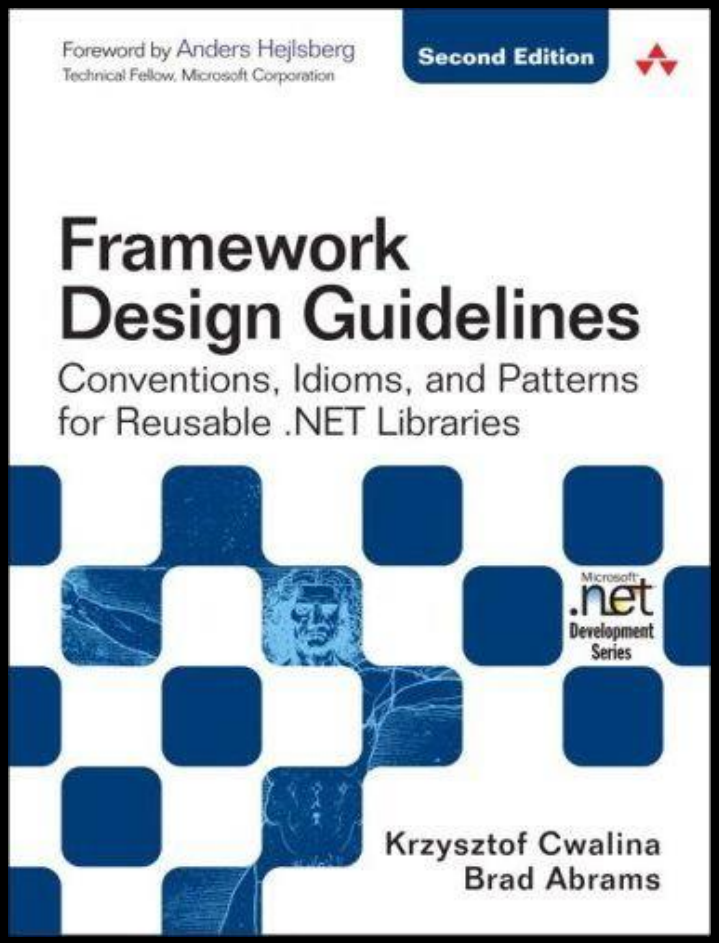
© Theo Jungeblut 2014

THE "MUST READ"-BOOK(S)

by Krzysztof Cwalina, Brad Abrams

Framework Design Guidelines

"teaches developers the best practices for designing reusable libraries for the Microsoft .NET Framework."



© Theo Jungeblut 2014

SUMMARY CRAFTSMANSHIP

Maintainability is achieved through:

- Simplification and Specialization *(KISS, SoC, SRP, OCP,)*
- Readability
  - Coding Guidelines
  - Naming
  - Code Structure
  - Formating
  - Doing Comments right!

© Theo Jungeblut 2014

SOFTWARE CRAFTSMANSHIP

Next Up: Lab