South Bay

Clean Code

Why Clean Code matters

Mountain View, March 2nd 2011

Theo Jungeblut

- Senior Software Developer at Omnicell Inc. in Mountain View
- Has been designing and implementing .NET based applications , components and frameworks for 8 years
- Previously worked in factory automation with focus on component based software and framework development for 3 ½ years
- Degree in Software Engineering and Network Communications

theo@csharp-lighthouse.com

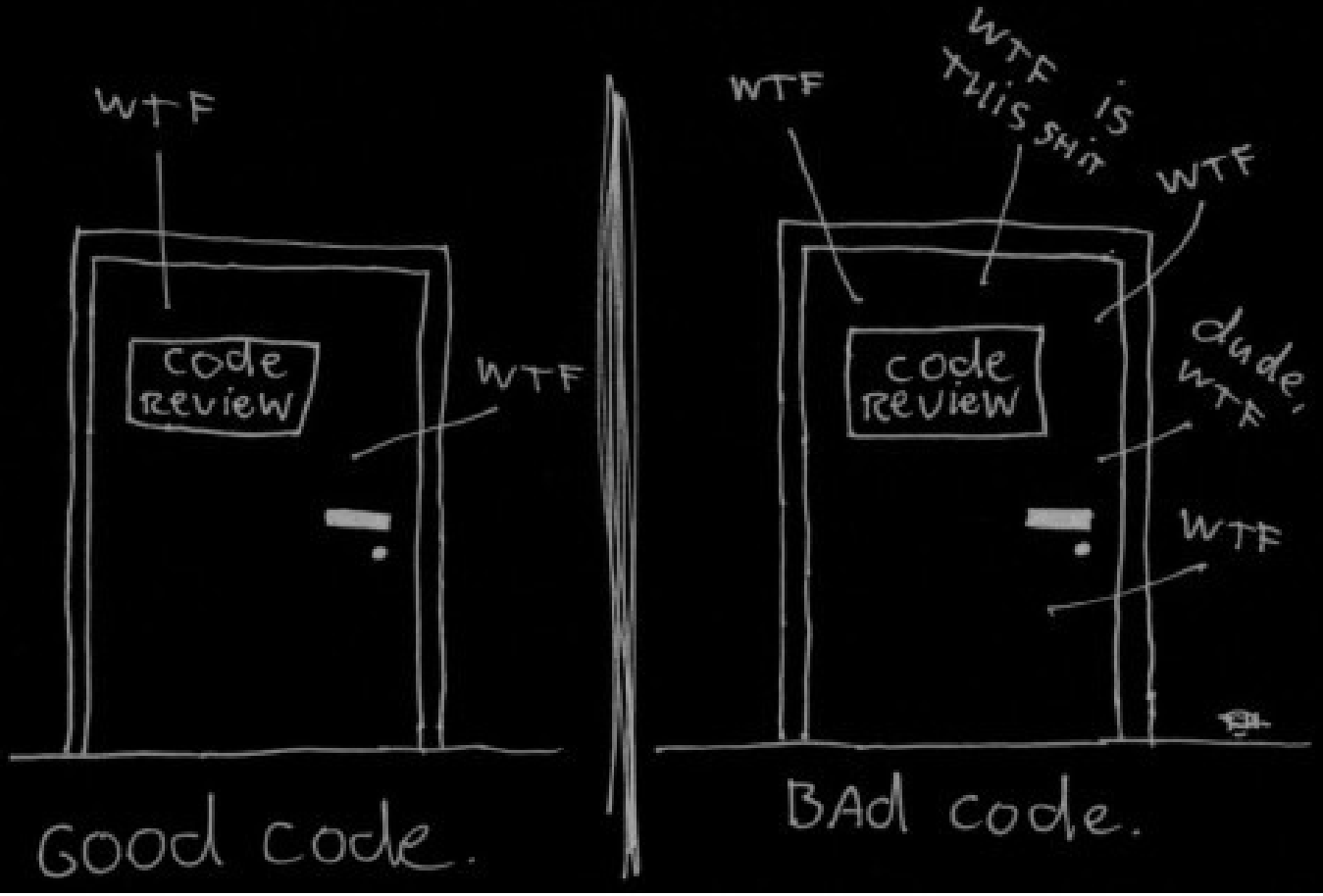
www.csharp-lightouse.com

Overview

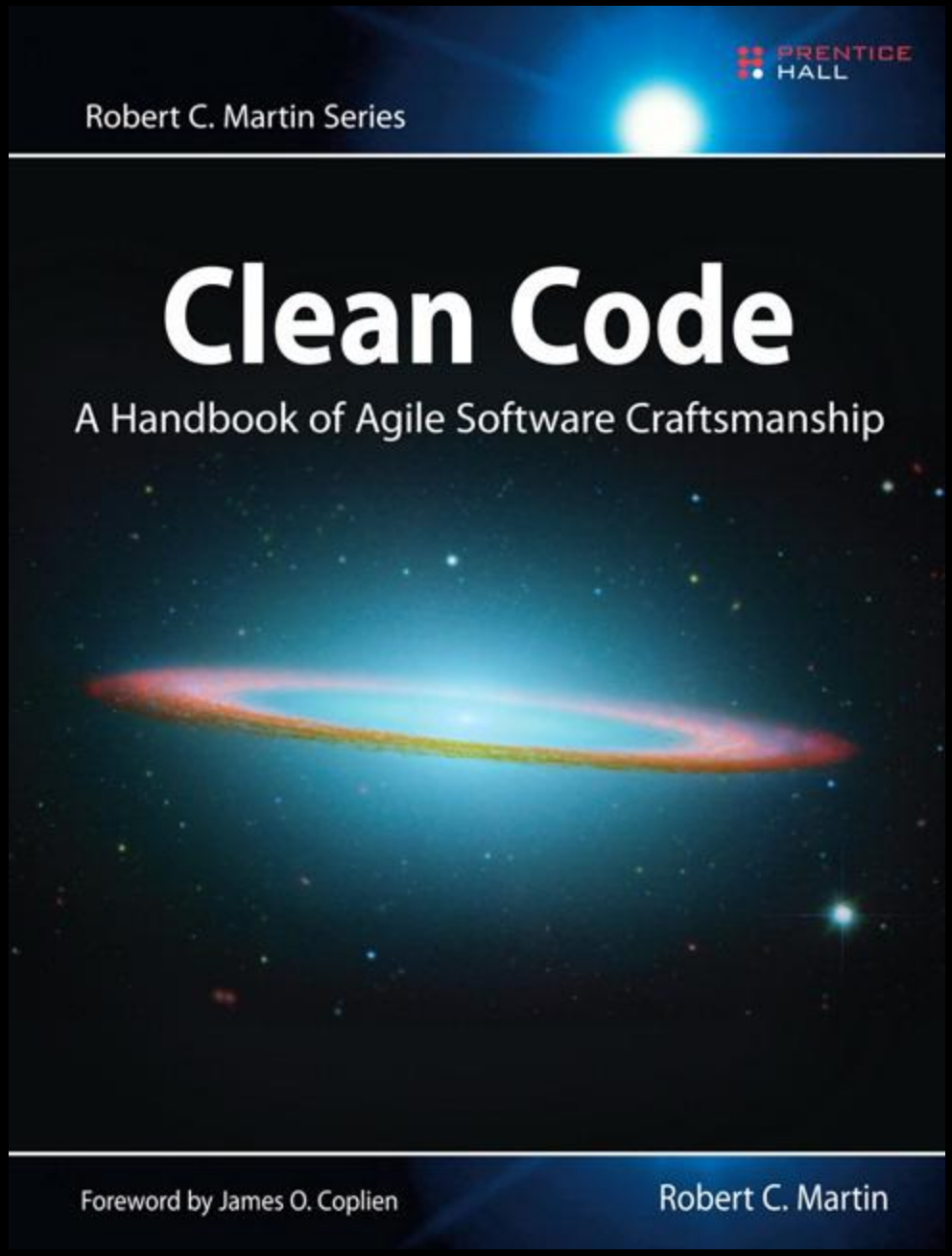
- Why Clean Code
- Tools
  - Resharper
  - FxCop, StyleCop & StyleCop plugin for Resharper
  - GhostDoc & Spell Checker
  - Code Contracts, Pex & Moles
- Clean Code Developer Initiative
- Principles and Practices
- Code Comparison
- Q&A

Does writing Clean Code make us more efficient?

The ONLY VALID measurement of code QUALITY: WTFs/minute

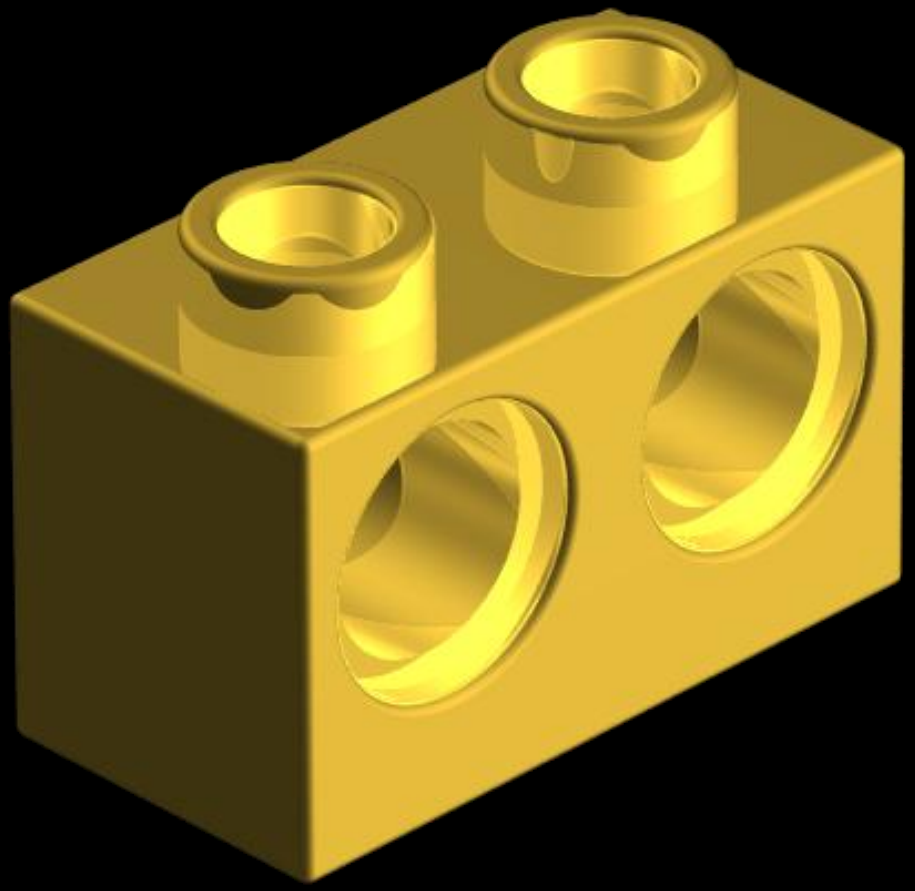


What is Clean Code?


PRENTICE HALL
Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin

http://www.technicopedia.com/8865.html

http://www.technicopedia.com/8865.html



http://technicbricks.blogspot.com/2009/06/tbs-techpoll-12-results-2009-1st.html

.NET Tools and their Impact

Tool name	Positive Impact	Negative Impact
Resharper	compiling -- -- --	VS responsiveness --
FxCop	code quality +++	compiling +
StyleCop	code quality	compiling +
StyleCop plugin for Resharper	compiling ---	VS responsiveness --
Ghost Doc	automated docs	potentially worse doc
Spell Checker	fewer spelling errors	performance --
Code Contracts	testability, quality +++	compiling ++
Pex & Moles	automated test	limited

Resharper

"The single most impacting development addition to Visual Studio"

Features:

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

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

FxCop / Code Analysis

Static Code Analysis:

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

- Design-by-Contract programming
- Improved testability
- Static verification
- API documentation

http://msdn.microsoft.com/en-us/devlabs/dd491992

Method Overview

Method contracts should be written with the different elements ordered as follows:

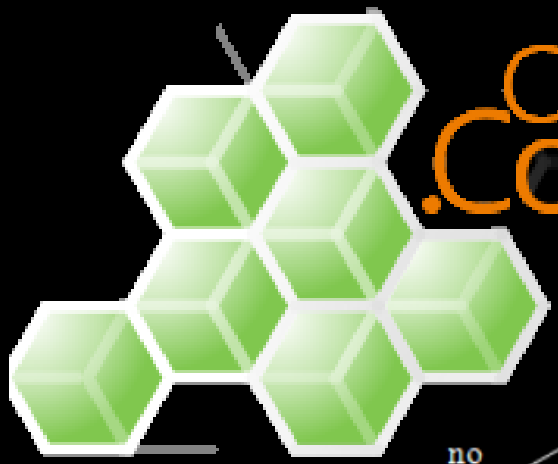
If-then-throw	Backward-compatible public preconditions
Requires, Requires⟨E⟩	All public preconditions
Ensures	All public (normal) postconditions
EnsuresOnThrow	All public exceptional postconditions
Ensures	All private/internal (normal) postconditions
EnsuresOnThrow	All private/internal exceptional postconditions
EndContractBlock	If using if-then-throw-style preconditions without any other contracts, place a call to EndContractBlock to indicate all previous if checks are preconditions.

Code Contracts User Manual, 5.3 Contract Ordering, page 22

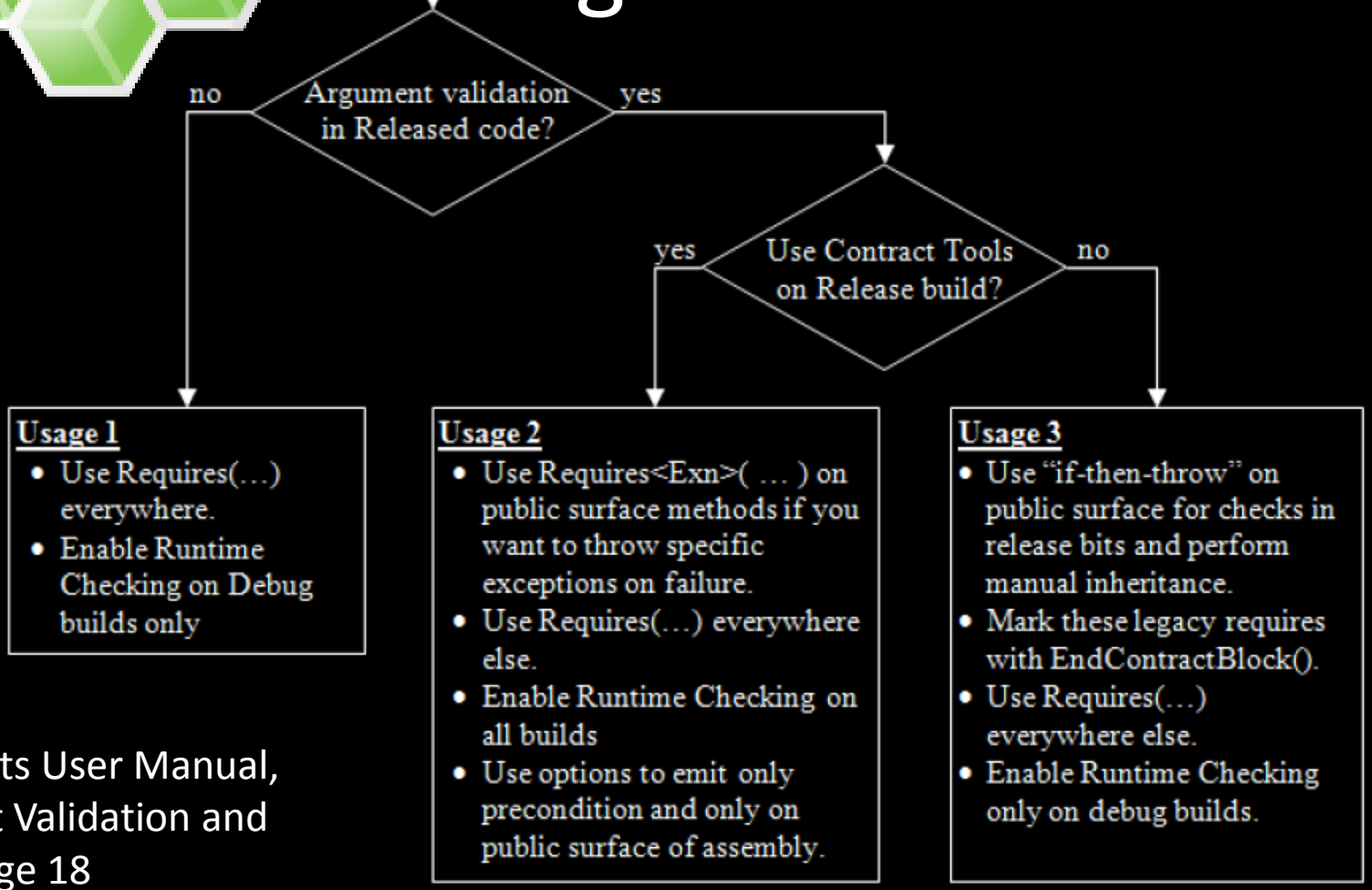
Runtime Checking Levels

Checking Level	Enabled Runtime Checks						
	Legacy	Requires⟨E⟩	Requires	Ensures	Invariants	Asserts	Assumes
Full	X	X	X	X	X	X	X
Pre and Post	X	X	X	X			
Preconditions	X	X	X				
ReleaseRequires	X	X					
None							

Code Contracts User Manual, 6.2.1 Runtime Checking Level, page 24



Argument Validation

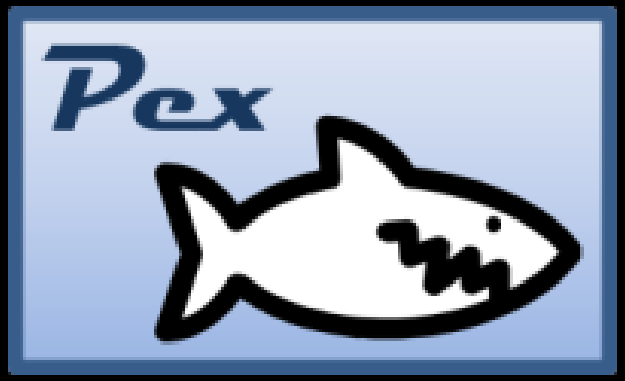


Code Contracts User Manual, 5.1 Argument Validation and Contracts, page 18

Shortcut	Contract Snippet
cr	Contract.Requires(...);
ce	Contract.Ensures(...);
ci	Contract.Invariant(...);
crr	Contract.Result⟨...⟩()
co	Contract.OldValue(...)

cim	[ContractInvariantMethod]
	private ObjectInvariant() {
	  Contract.Invariant(...);
	}

crn	Contract.Requires(... != null);
cen	Contract.Ensures(Contracts.Result⟨...⟩() != null);
crsn	Contract.Requires(!String.IsNullOrEmpty(...));
cesn	Contract.Ensures(!String.IsNullOrEmpty(Contracts.Result⟨string⟩()));
cca	Contract.Assert(...);
cam	Contract.Assume(...);
cre	Contract.Requires⟨E⟩(...);
cren	Contract.Requires⟨ArgumentNullException⟩(... != null);
cresn	Contract.Requires⟨ArgumentException⟩(!String.IsNullOrEmpty(...));
cintf	expands to an interface template and associated contract class



Microsoft Pex & Moles

- Pex automatically generates test suites with high code coverage.
- Moles allows to replace any .NET method with a delegate.

http://research.microsoft.com/en-us/projects/pex/

Ghost Doc

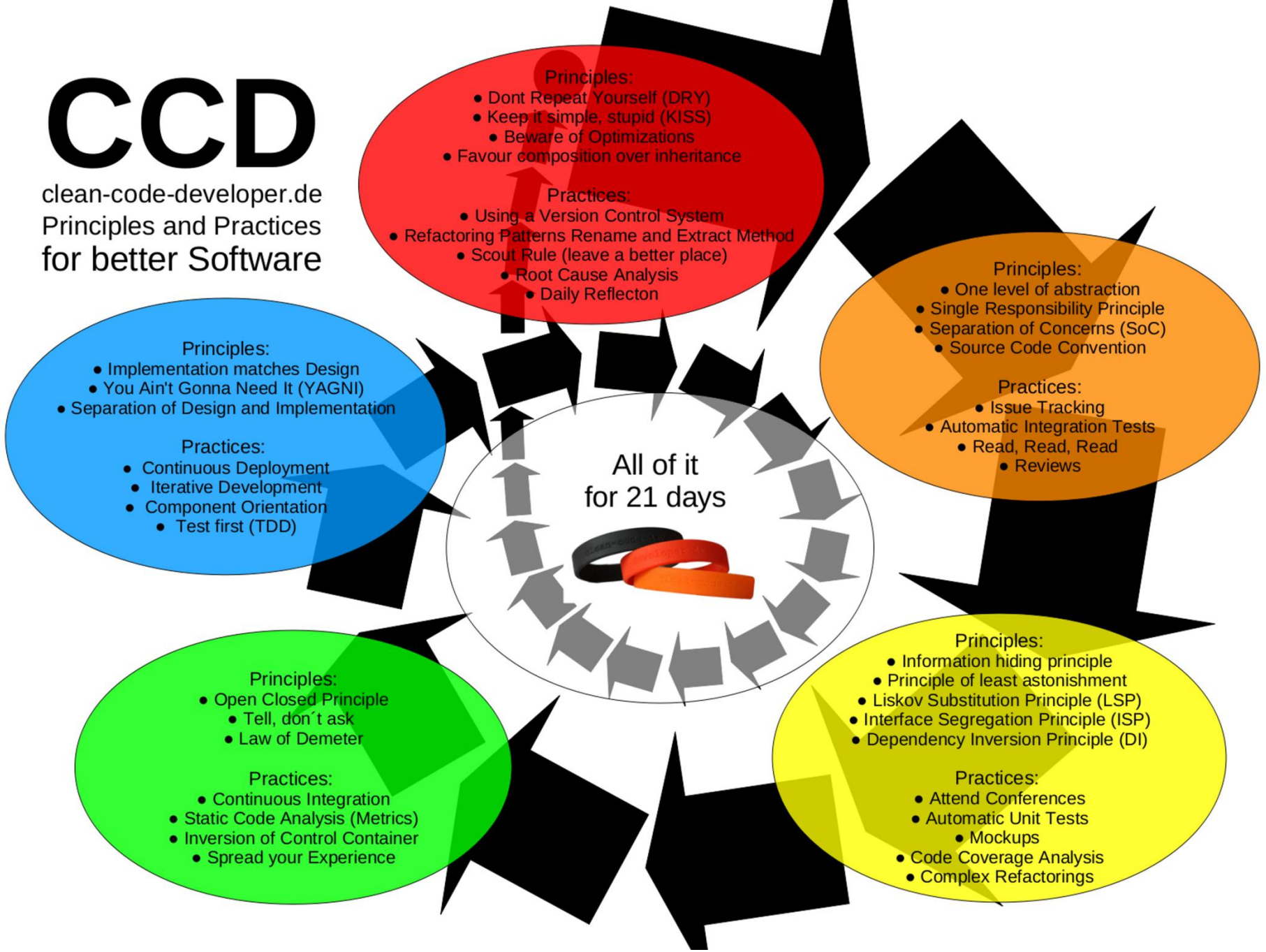
- Save keystrokes and time
- Simplify documenting your code
- Benefit of the base class documentation

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

Spell Checker

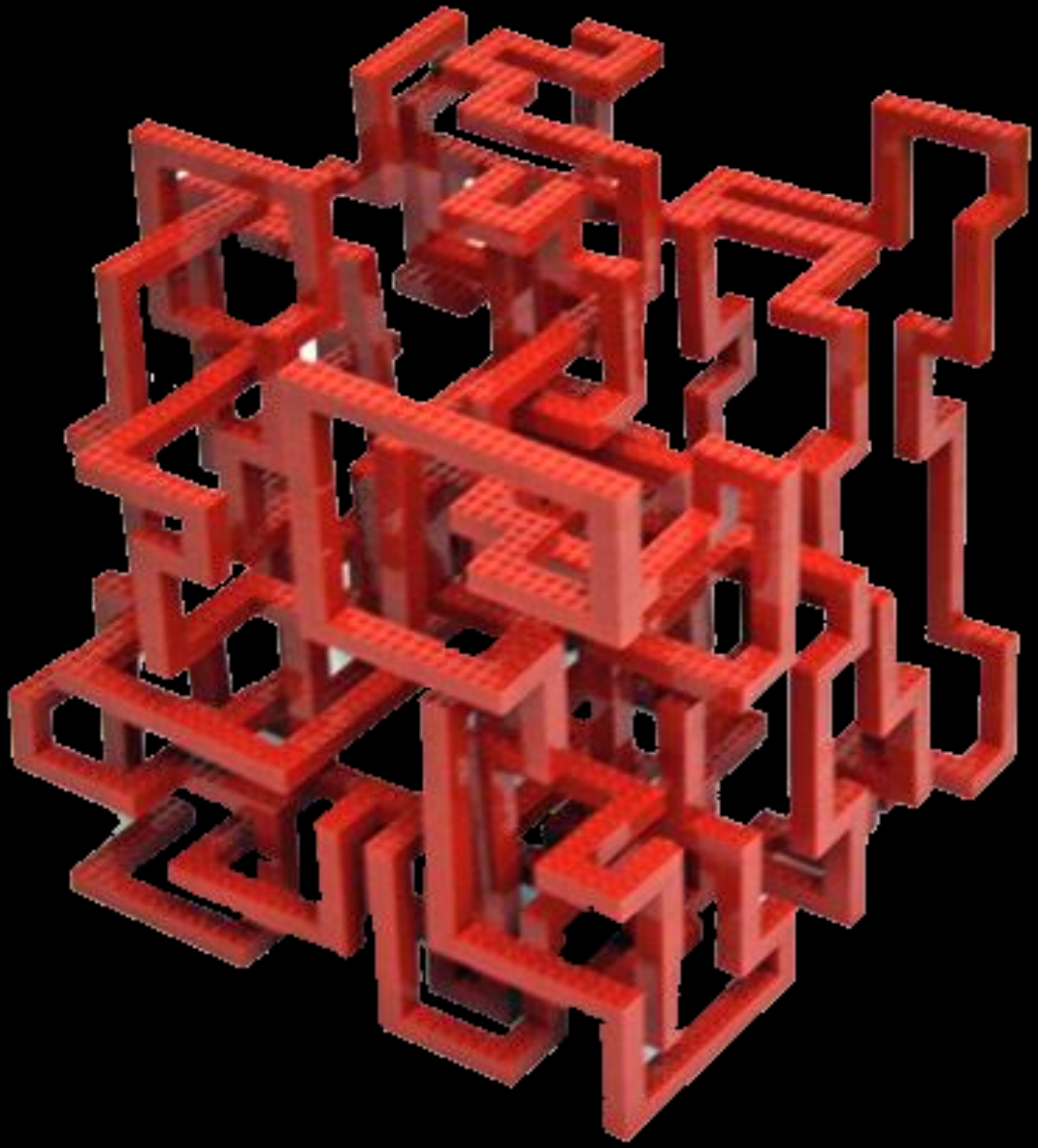
- Spelll chicking for texts and comments in VS

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



Graphic by Michael Hönnig http://michael.hoennig.de/2009/08/08/clean-code-developer-ccd/

Keep it simple, stupid
(KISS)



Graphic by Nathan Sawaya courtesy of brickartist.com

Don’t repeat yourself (DRY)

Separation of Concerns (SoC)

Single Responsibility Principle (SRP)

Component-Oriented Programming (CoP)

Dependency Inversion Principle (DIP)

S

O

L

I

D

Single Responsibility Principle

Open/Closed Principle

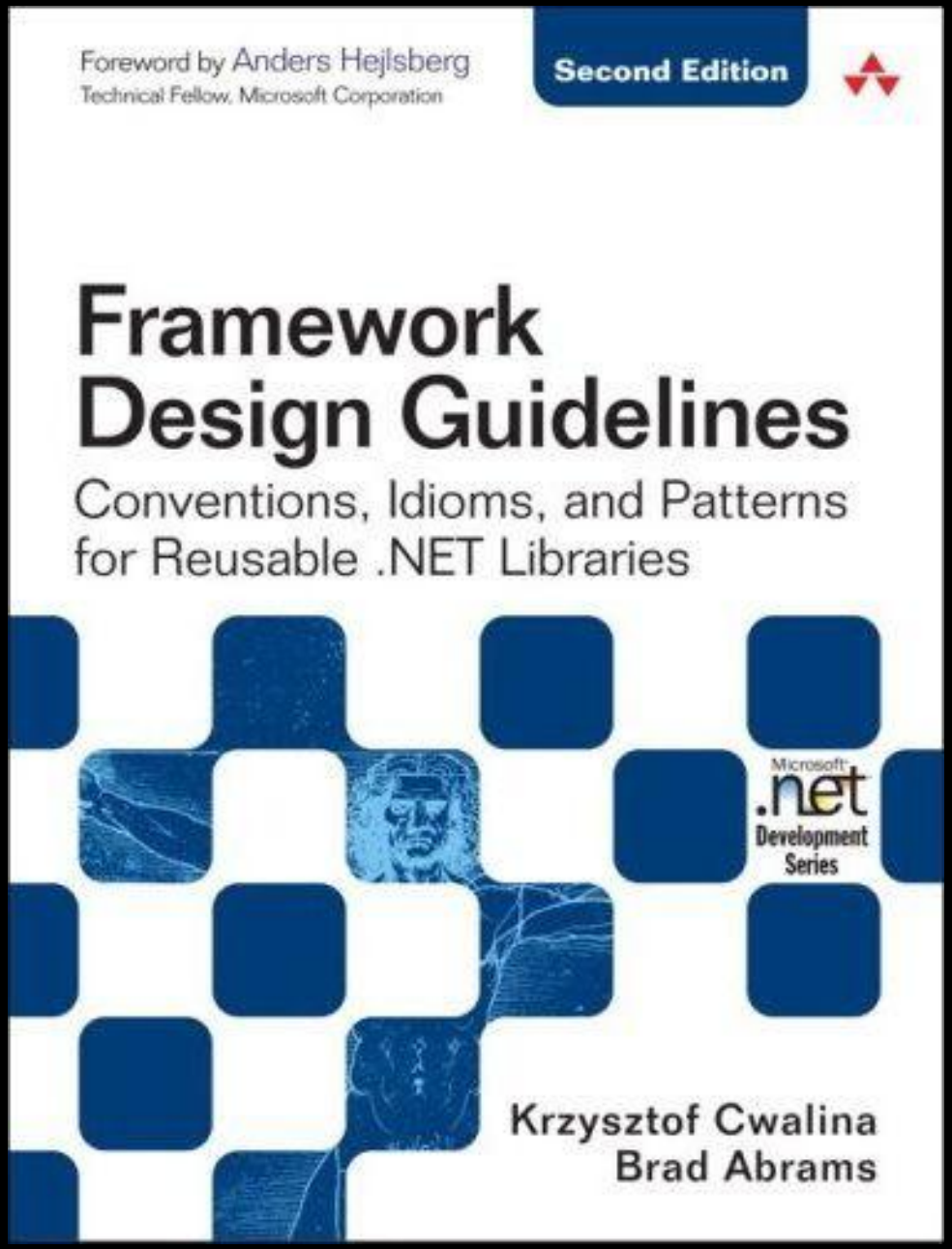
Liskov Substitution Principle

Interface Segregation Principle

Dependency Inversion Principle

Robert C Martin: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

Source Code Conventions

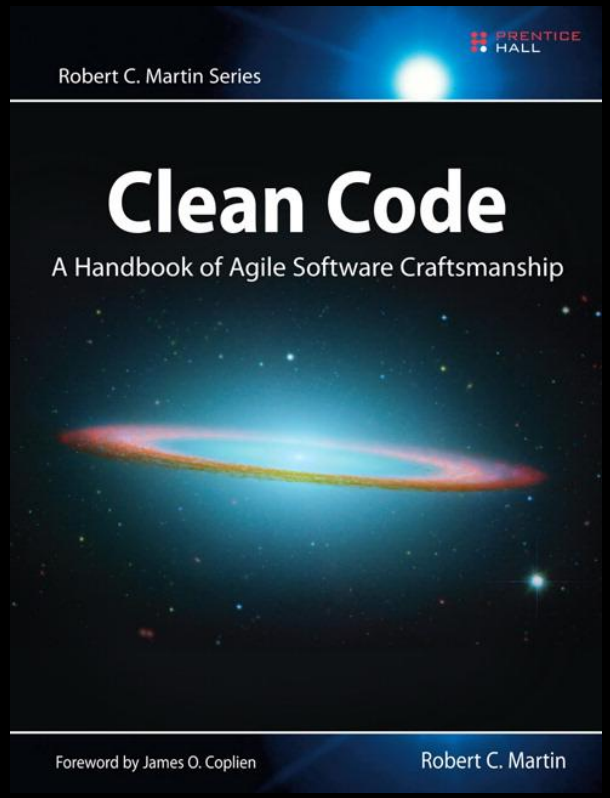

Foreword by Anders Hejlsberg
Technical Fellow, Microsoft Corporation
Second Edition
Framework Design Guidelines
Conventions, Idioms, and Patterns for Reusable .NET Libraries
Microsoft .net Development Series
Krzysztof Cwalina
Brad Abrams

Let’s switch to the code

Summary Clean Code

Being more efficient through:

- Simplification and Specialization
  (KISS, SoC, SRP)
- Decoupling
  (Contracts, CoP, IoC or SOA)
- Avoiding Code Blow (DRY, YAGNI)
- Quality through Testability
  (all of them!)



Q & A

Graphic by Nathan Sawaya courtesy of brickartist.com

References

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

FxCop / Code Analysis
http://msdn.microsoft.com/en-us/library/bb429476(VS.80).aspx
http://blogs.msdn.com/b/codeanalysis/
http://www.binarycoder.net/fxcop/index.html

Code Contracts
http://msdn.microsoft.com/en-us/devlabs/dd491992
http://research.microsoft.com/en-us/projects/contracts/

Pex & Mole
http://research.microsoft.com/en-us/projects/pex/

StyleCop
http://stylecop.codeplex.com/

StyleCop Plugin for Resharper
http://stylecopforresharper.codeplex.com/

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

Lego (trademarked in capitals as LEGO)

goto

www.csharp-lighthouse.com

for

slides & code examples