Code Quality Assurance
(summary of all presentations)

Peter Kofler, ‘Code Cop’
IBM, March 2012

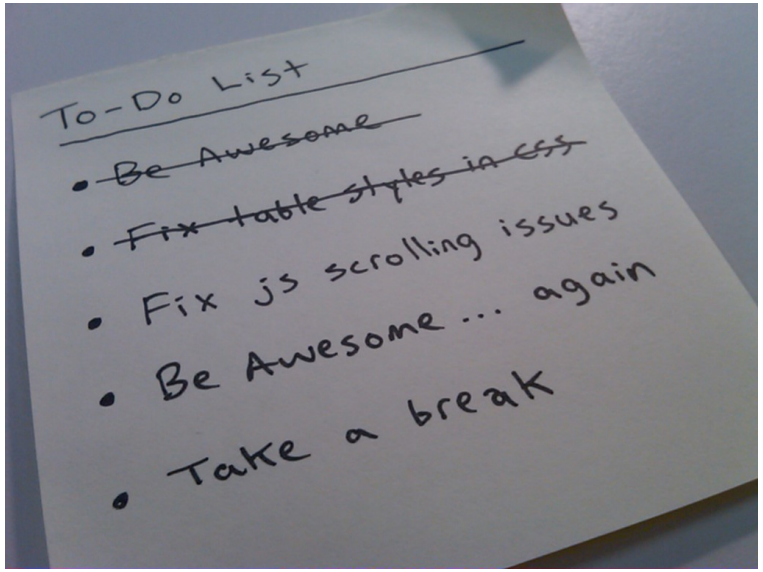

562
564
NAL DISEÑO

Peter Kofler

- Ph.D. (Appl. Math.)
- Professional Software Developer for 13 years
- Lead Developer at IBM
- “fanatic about code quality”

The opinions expressed here are my own and do not necessarily represent those of current or past employers.

Tools of the Trade



Mindset/Books

Code Kata

TDD/BDD

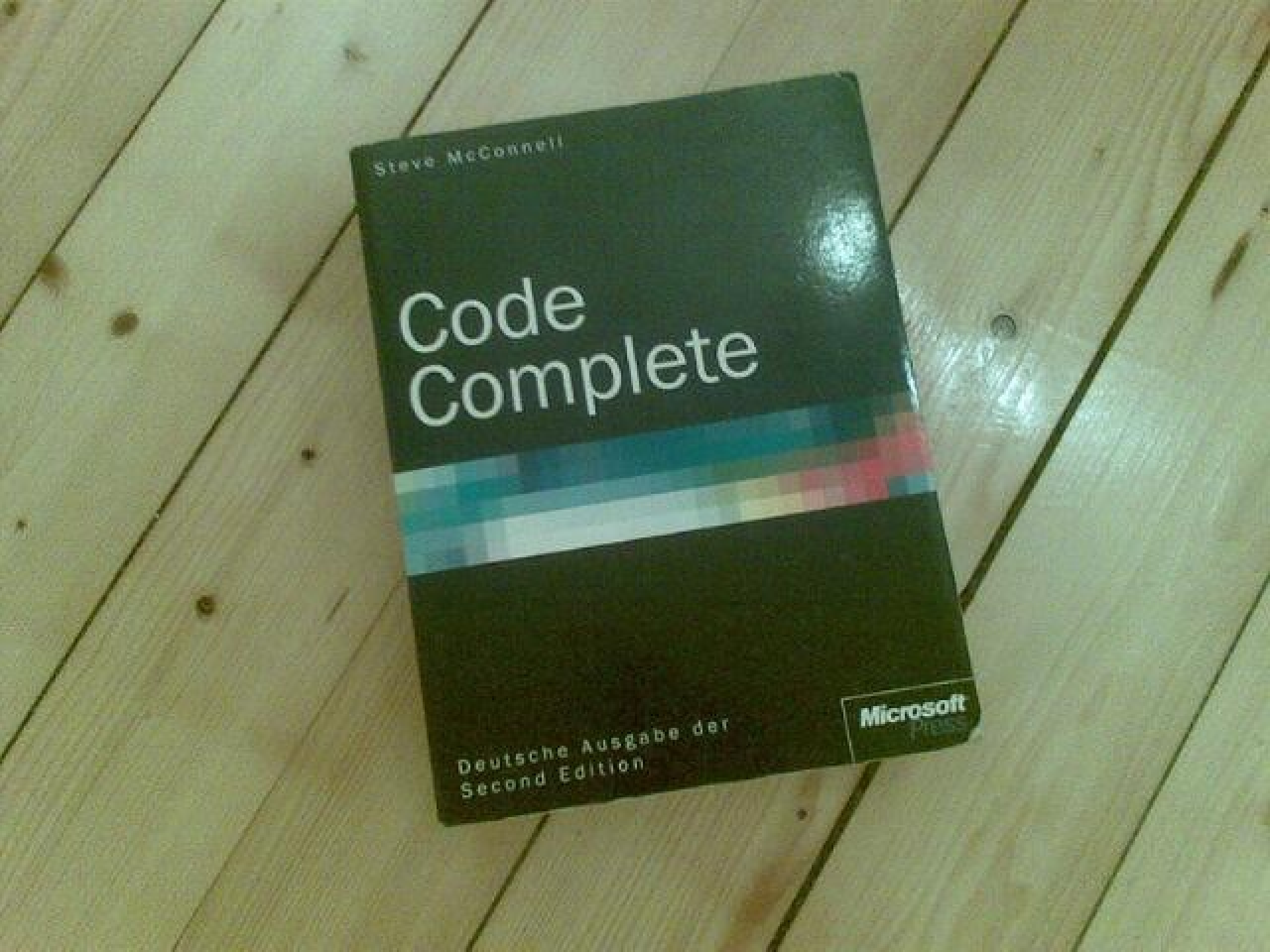
Code Coverage

Continuous Integration

Static Analysis

Code Review

Zero-Defect Mindset (MSF)

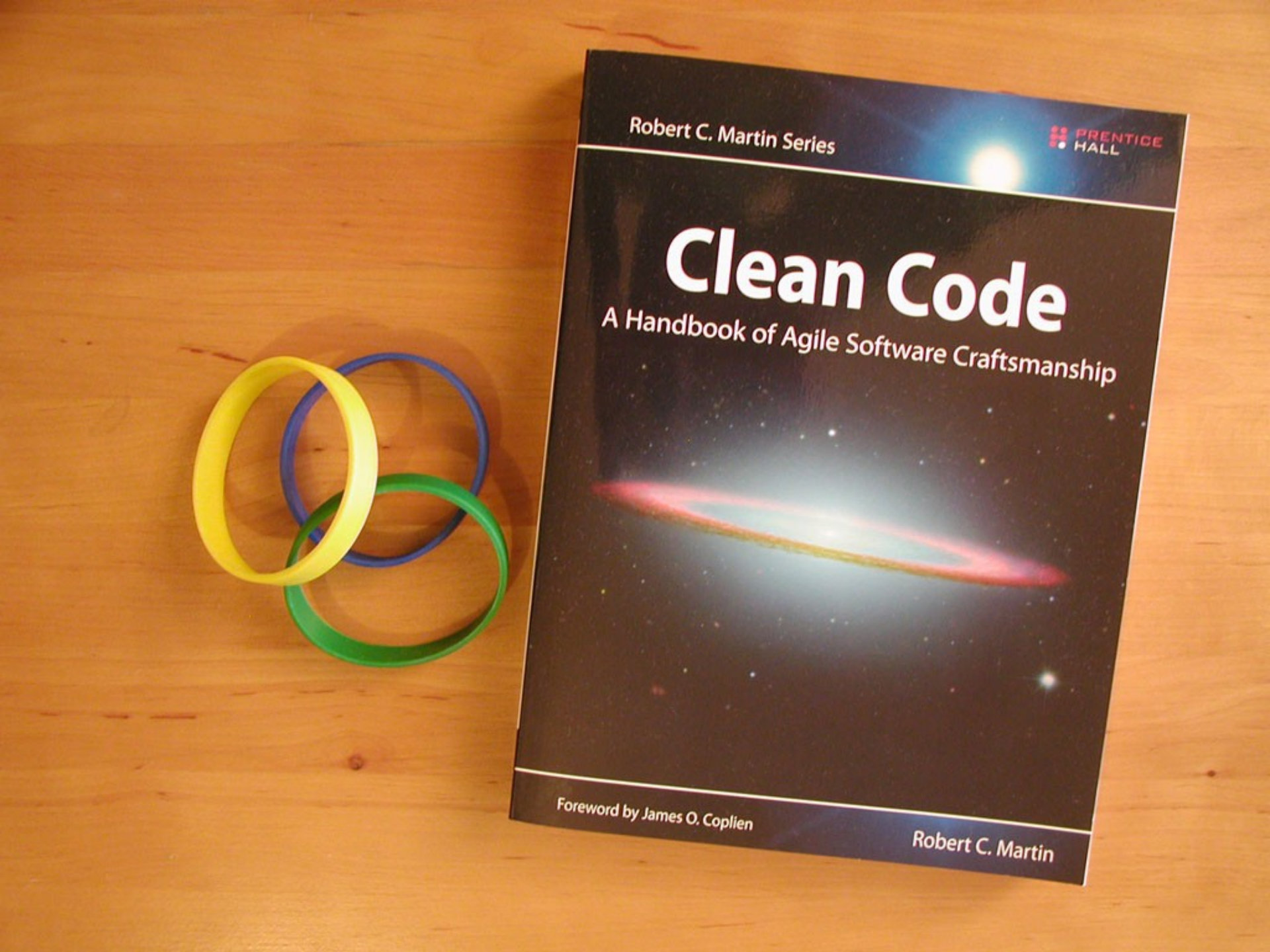

Steve McConnell
Code Complete
Microsoft PRESS
Deutsche Ausgabe der
Second Edition


GET
PAID!
Inc
Oliver
CROPS
SHL
epc
yeah

Boy Scout Rule


b-agile
Andrew Hunt
David Thomas
Mit einem Vorwort von
Ward Cunningham
Der Pragmatische
Programmierer
Übersetzt von Andreas Braig und Steffen Gemkow
HANSER

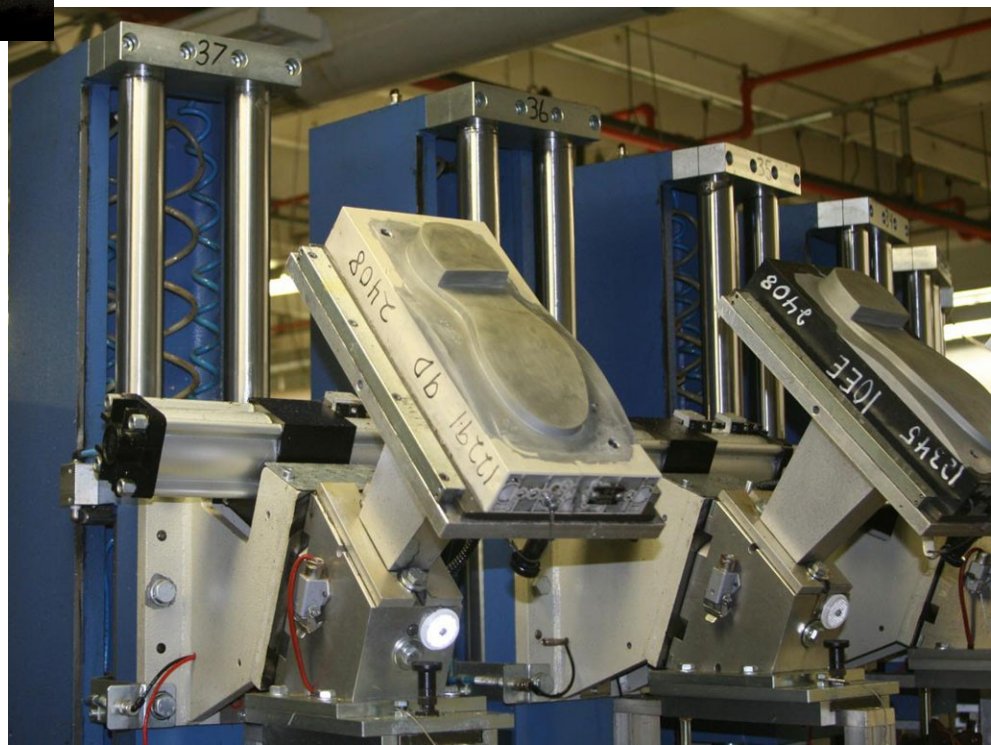
Robert C. Martin (“Uncle Bob”)


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

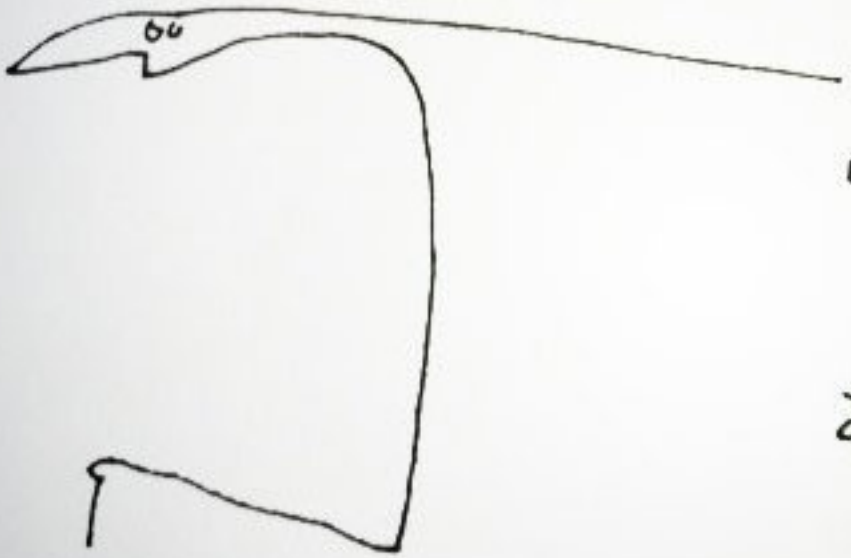
Software Craftsmanship

Craftsmanship

Engineering



Craftsmanship Over Crap



suddenly
"passion"
became
a buzzword!

Mindset

and

Passion

The Prime Factors Kata

The Requirements.

- Write a class named “PrimeFactors” that has one static method: generate.
  - The generate method takes an integer argument and returns a List<Integer>.
  - That list contains the prime factors in numerical sequence.

http://butunclebob.com/ArticleS.UncleBob.ThePrimeFactorsKata

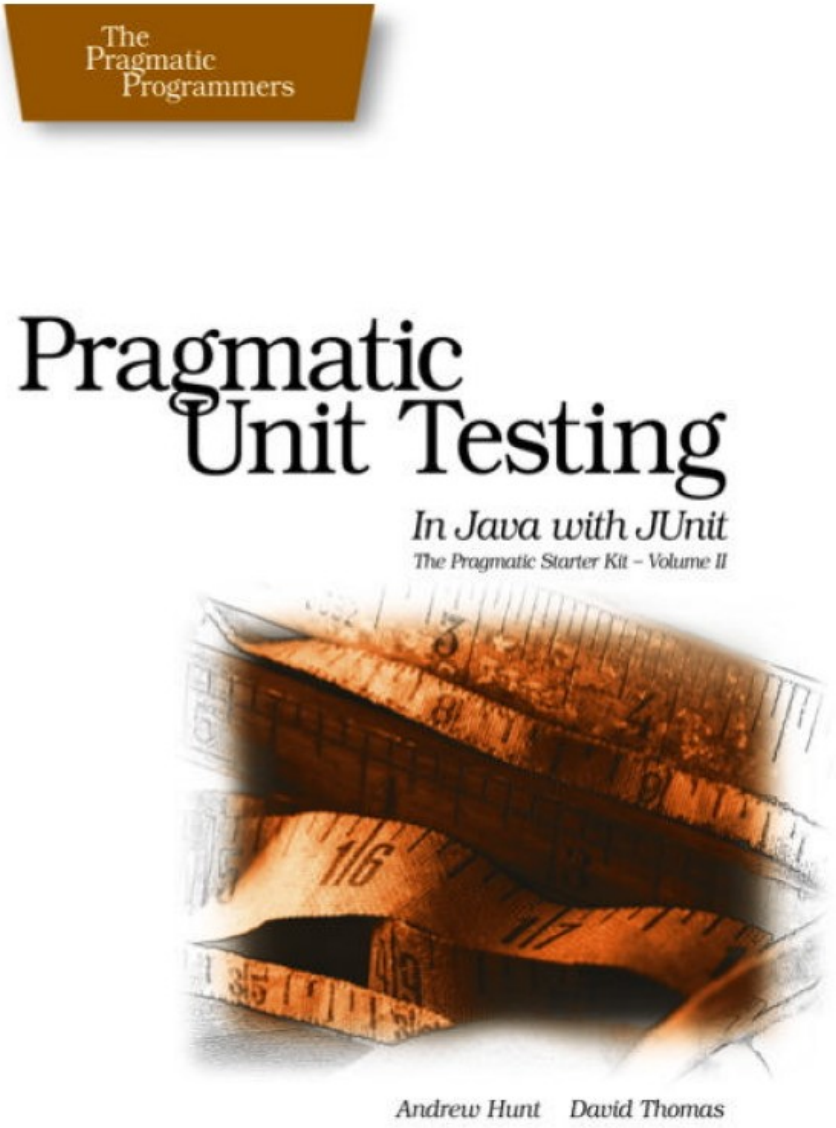
Demo

Keep the bar green to keep the code clean

Unit Testing

- test individual units
- isolate each part
- show that the individual parts are correct
- regression testing
- sort of living documentation
- executed within a framework

http://en.wikipedia.org/wiki/Unit_testing


The Pragmatic Programmers
Pragmatic Unit Testing
In Java with JUnit
The Pragmatic Starter Kit – Volume II
Andrew Hunt David Thomas

xUnit Frameworks

- e.g.
- Smalltalk - SUnit
- Java - JUnit, TestNG
- .NET - NUnit, Typemock
- C++ - CppUTest
- Groovy, PHP, Python, Ruby, Scala, etc.

http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

Test-Driven Development

- add a test
- run all tests and see if the new one fails
- write some code
- run all tests and see them succeed
- refactor code
- „Red Green Refactor“

http://en.wikipedia.org/wiki/Test_Driven_Development

A minute ago all their code worked


Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Entwurfsmuster
Elemente wiederverwendbarer objektorientierter Software
ADDISON-WESLEY
The Addison-Wesley Signature Series
A Kent Beck Signature Book
Test-Driven Development
By Example
Kent Beck
Refacto
Wie Sie das Design vorhandener Softw verbessern
Erich Gamma
119

BDD
is
TDD
“Done Right“

Scenarios and Stories

- focus on behaviour
  (e.g. requirements are behaviour)
- full sentences
- expected behaviour **should**
- Use Arrange-Act-Assert in form of
  Given-When-Then

The Prime Factors Kata (again)

Demo

Testing is
a mindset

You

have to want it


مكتبة الإيمان
الجمعية الشرعية للتقوية
الرياضيات
التاريخ

Code Coverage

comprehensiveness of tests

Beware!

comprehensiveness ≠ quality!

100%

(What else?)

Some Code Coverage Tools

Java - EMMA, Cobertura, Clover

.NET - NCover, PartCover

C++ - Covtool, gcov

Ruby - rcov

commercial products

Demo

Demo


works on
my
machine

Daily Builds
Nightly Builds

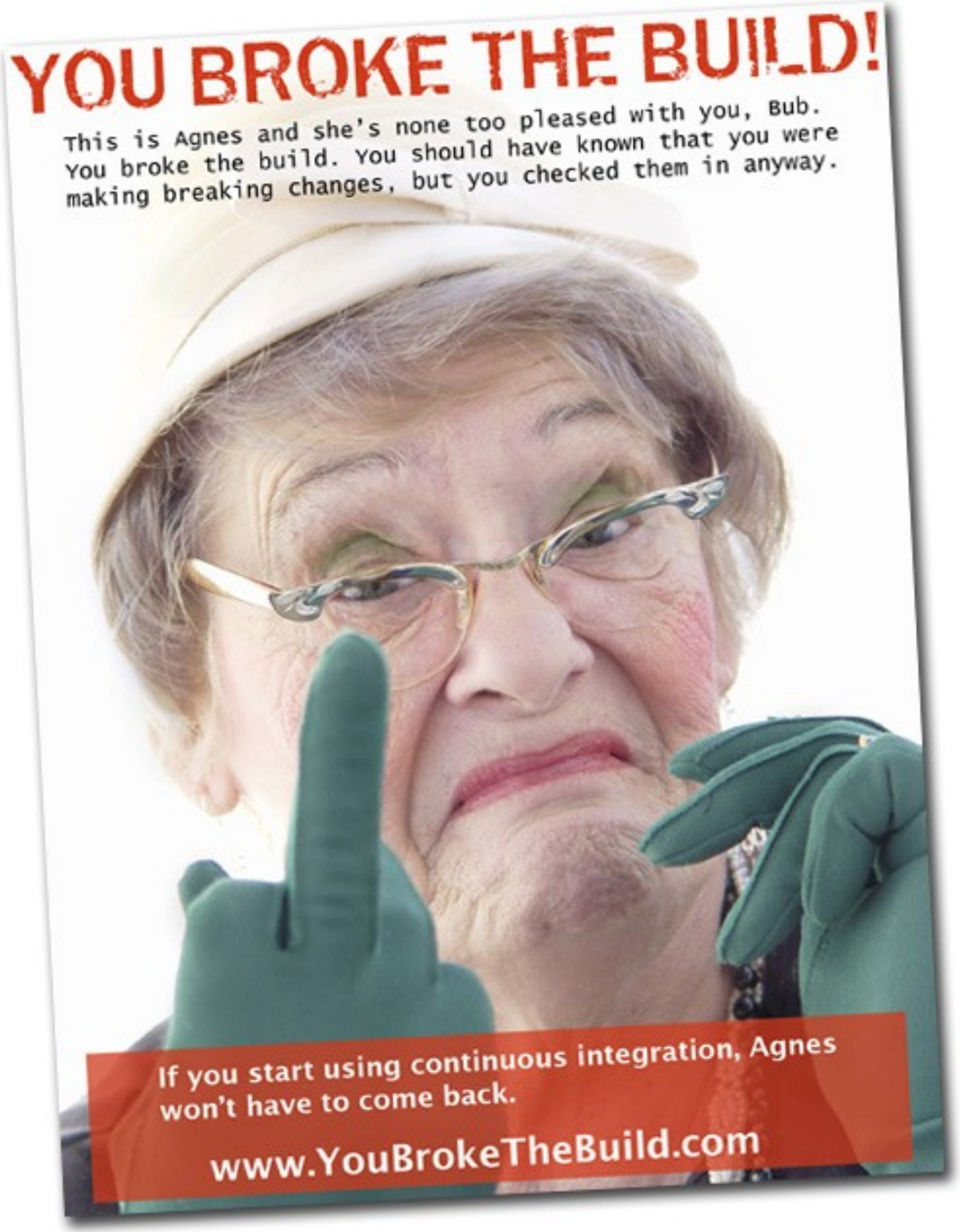
Continuous Integration

Continuous Integration

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits every day
- Every commit should be built
- Keep the build fast
- Everyone can see the results of the build

http://en.wikipedia.org/wiki/Continuous_integration

Immediate Feedback and Early Warning


YOU BROKE THE BUILD!
This is Agnes and she's none too pleased with you, Bub. You broke the build. You should have known that you were making breaking changes, but you checked them in anyway.
If you start using continuous integration, Agnes won't have to come back.
www.YouBrokeTheBuild.com

Some CI Servers

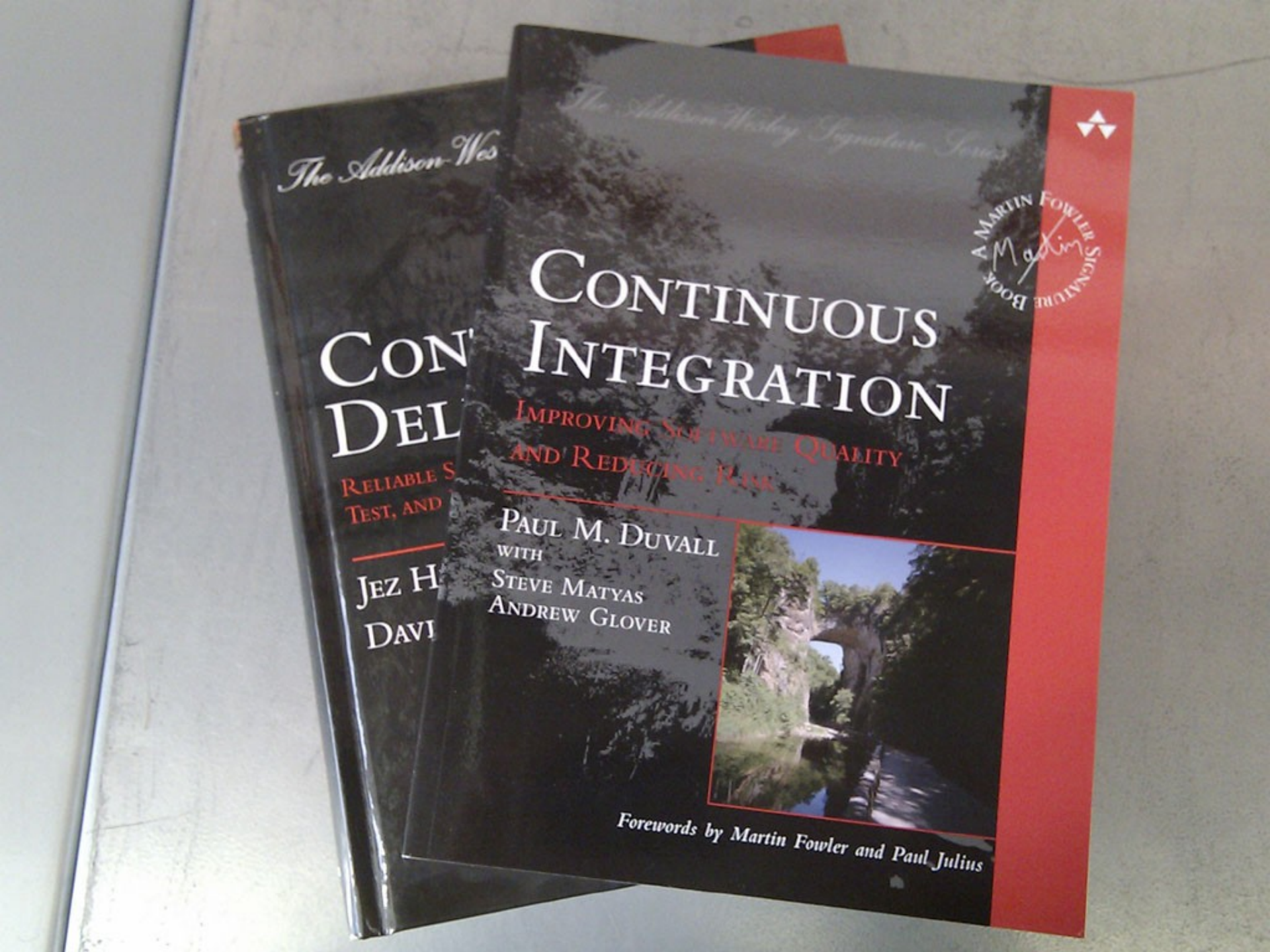
Java - CruiseControl, Jenkins (Hudson)

.NET - TFS, CruiseControl.NET, Jenkins

C++ - Jenkins (Shell/Make)

Ruby - CruiseControl.rb

commercial products - TeamCity, ALA


The Addison-Wes
CON
DEL
RELIABLE S
TEST, AND
JEZ H
DAVI
The Addison-Wesley Signature Series
CONTINUOUS INTEGRATION
IMPROVING SOFTWARE QUALITY AND REDUCING RISK
PAUL M. DUVALL
WITH
STEVE MATYAS
ANDREW GLOVER
A MARTIN FOWLER SIGNATURE BOOK
Martin
Forewords by Martin Fowler and Paul Julius

Good Enough?

Demo


44 40 32
1/20

Check What?

- Lexical analysis
  - naming, coding conventions, design idioms
- Flow/path analysis
  - null-pointer, dead code (conditional)
- Dependency analysis
  - architectural/design flaws
- Verification
  - mathematical proof of correctness

Some Static Analysis Tools

Java - Checkstyle, Findbugs, PMD

.NET - FxCop, (ReSharper, NDepend)

C++ - (Sp)Lint

Ruby - Roodi, Dust, Flog, Saikuro

commercial products

Good Enough?

Demo

Some Code Review Tools

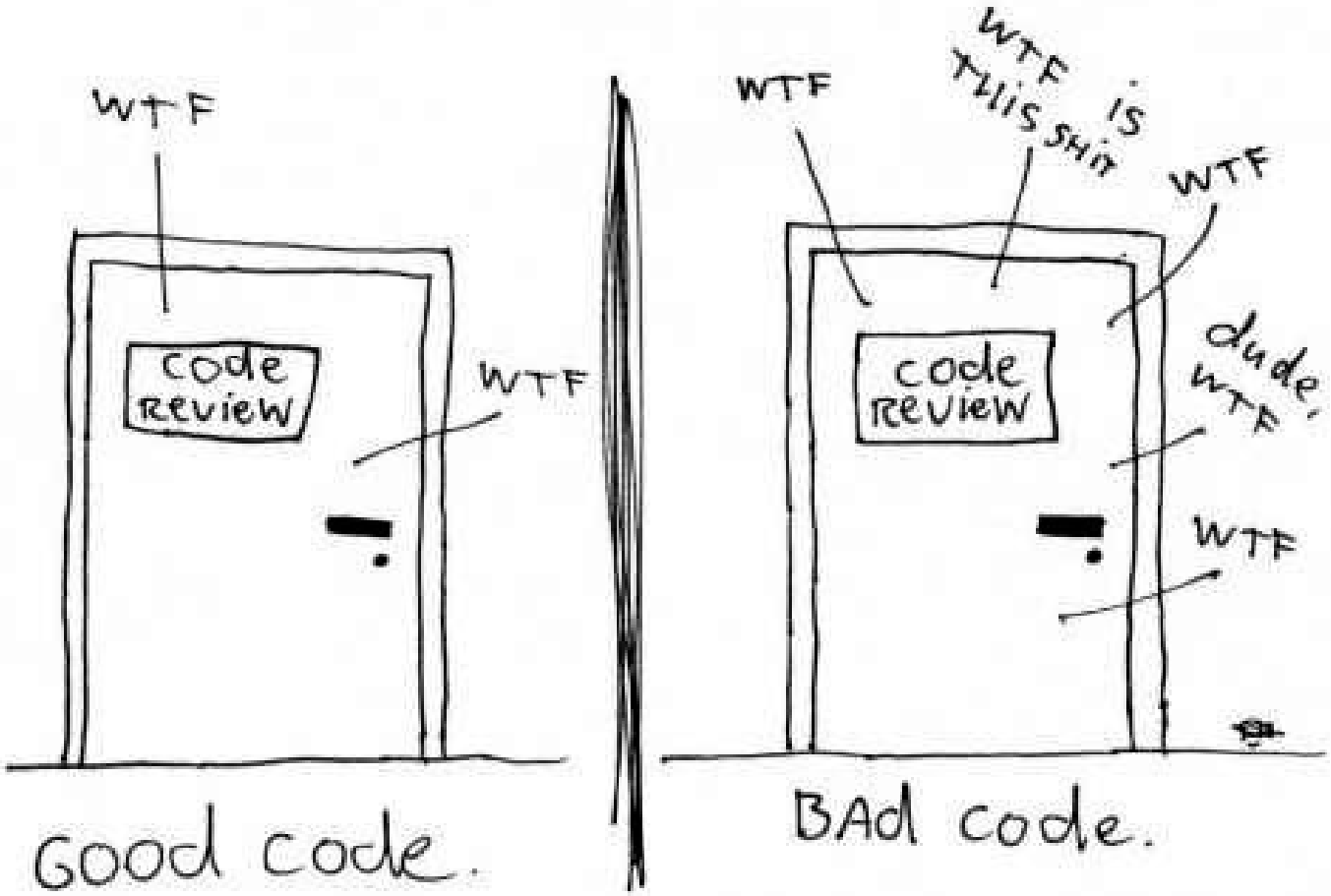
Your peer (Pair Programming)

Your version control system

Your IDE (e.g. ReviewClipse)

commercial products

The only valid measurement of code quality: WTFs/minute



Good Enough?

(Probably Not)

Lessons Learned

(!) Boy Scout Rule

(!) Do TDD/BDD

(!) Use the Tools

(Coverage, CI, Static Analysis)

Thank You

www.code-cop.org

@codecopkofler

Links #1

Mindset

- http://codebetter.com/blogs/darrell.norton/archive/2003/12/03/4222.aspx
- http://www.sqnz.org.nz/documents/ShipHappens/Software%20Quality%20Group%20Presentation_frame.htm (Zero-Defect Mindset)
- http://pragprog.com/the-pragmatic-programmer
- http://en.wikipedia.org/wiki/Fixing_Broken_Windows
- http://programmer.97things.oreilly.com/wiki/index.php/The_Boy_Scout_Rule

Software Craftsmanship

- http://en.wikipedia.org/wiki/Software_craftsmanship
- http://en.wikipedia.org/wiki/Robert_Cecil_Martin
- http://vikashazrati.wordpress.com/2009/11/18/dissecting-software-craftsmanship/
- http://clean-code-developer.de/

Links #2

Kata

- http://en.wikipedia.org/wiki/Kata_(programming)
- http://butunclebob.com/ArticleS.UncleBob.ThePrimeFactorsKata
- http://hg.code-cop.org/primefactors/ (Kata Source Code)

Unit Test

- http://en.wikipedia.org/wiki/Unit_testing
- http://en.wikipedia.org/wiki/XUnit
- http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

TDD

- http://en.wikipedia.org/wiki/Test_Driven_Development

BDD

- http://en.wikipedia.org/wiki/Behavior_Driven_Development
- http://dannorth.net/introducing-bdd/

Links #3

Code Coverage

- http://en.wikipedia.org/wiki/Code_coverage
- http://www.ibm.com/developerworks/java/library/j-cq01316/

Continuous Integration

- http://en.wikipedia.org/wiki/Continuous_integration
- http://www.codinghorror.com/blog/archives/000818.html
- http://www.stevemcconnell.com/ieeesoftware/bp04.htm
- http://www.joelonsoftware.com/articles/fog0000000023.html
- http://jenkins-ci.org/

Static Code Analysis

- http://en.wikipedia.org/wiki/Static_code_analysis
- http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

CC Images

- spray face: http://www.flickr.com/photos/iangallagher/4115047191/
- Judge Dredd: http://www.flickr.com/photos/eldave/6169431454/
- broken egg: http://www.flickr.com/photos/nickwheeleroz/2475011402/
- broken windows: http://www.flickr.com/photos/sketchglass/4281424410/
- shoe maker: http://www.flickr.com/photos/tbatty/1450209613/
- factory: http://www.flickr.com/photos/94693506@N00/4643248587/
- passion: http://gapingvoid.com/2011/03/29/suddenly-passion/
- red-green: http://www.flickr.com/photos/30830597@N08/3630649274/
- Darth Vader: http://rubystammtisch.at/
- covered car: http://www.flickr.com/photos/paulk/3166328163/
- works: http://www.codinghorror.com/blog/archives/000818.html
- Agnes: http://www.youbrokethebuild.com/
- microscope: http://www.flickr.com/photos/gonzales2010/8632116/
- questions: http://www.flickr.com/photos/seandreilinger/2326448445/