

Code Quality Assurance

FH Technikum Wien, April 2013

Peter Kofler, ‘Code Cop’
@codecopkofler
www.code-cop.org

• 562 •

• 564 •



AL DISEÑO

Peter Kofler

- Ph.D. (Appl. Math.)
- Professional Software Developer for 14 years
- Lead Developer at IBM
- “fanatic about code quality”

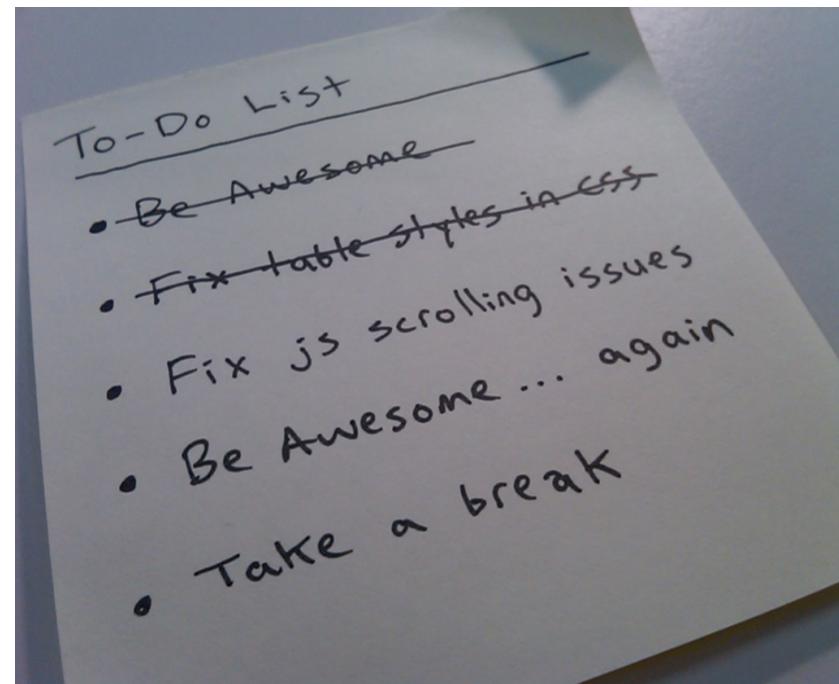


The opinions expressed here are my own and do not necessarily represent those of current or past employers.



Tools of the Trade

- Mindset/Books
- Code Kata
- TDD/BDD
- Code Coverage
- CI
- Static Analysis
- Code Review





Zero-Defect Mindset (MSF)

Steve McConnell

Code Complete

Deutsche Ausgabe der
Second Edition

Microsoft



Boy Scout Rule

btp-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 Series

PRENTICE
HALL

Clean Code

A Handbook of Agile Software Craftsmanship

Foreword by James O. Coplien

Robert C. Martin

Software Craftsmanship

FANATIC ABOUT CODE QUALITY

Craftsmanship



tboffy2007

Engineering



“Craftsmanship
Over Crap“
(Uncle Bob)



suddenly
"passion"
became
a buzzword!

Mindset

and

Passion

The Prime Factors Kata

What Exactly Will We Do?

- write code together
- using TDD
- see techniques
and patterns
- discuss while
doing



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>

First Some Math.

- **Prime Number:** number > 1 that has no divisors other than 1 and itself.
 - e.g. 2, 3, 5, 61, 67, ..., 997, ..., $2^{43112609}-1$
- **Prime Factors:** prime numbers that divide an integer without remainder.
 - e.g. $2 = 2,$
 $4 = 2 * 2,$
 $24 = 2 * 2 * 2 * 3$
 $288 = 2^5 * 3^2$

Requirements (for Mortals).

- code a file/module/class “PrimeFactors”
- function/method/routine “generate”
- accept an integer number as parameter
- return the prime factors of that number

Demo

(Code Kata)

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

The
Pragmatic
Programmers

Pragmatic Unit Testing

In Java with JUnit

The Pragmatic Starter Kit - Volume II



Andrew Hunt David Thomas

Some xUnit Frameworks

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

“I write unit tests for one reason: so my coworkers don't f*** up my code.”

(David Angry)

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 **mercilessly**
- „Red Green Refactor“

A minute ago all
their code worked



Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Entwurfsmuster

Elemente wiederverwendbarer
objektorientierter Software

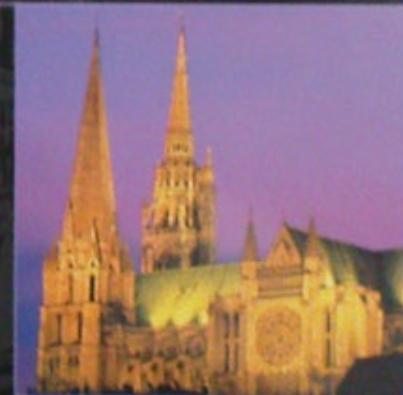
ADDISON-WESLEY

The Addison Wesley Signature Series

TEST-DRIVEN DEVELOPMENT

BY EXAMPLE

KENT BECK



A KENT BECK
SIGNATURE
BOOK

Refacto

Wie Sie das Design
vorhandener Softw-

er bessern

Frage von Kent Beck,
Bill William Opdyke,
Erich Gamma

119

ISBN 3-427-21670-6
Werkstatt 2. Auflage
EUROPAVERLAG AG

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

Testing is
a mindset

You
have to want it



TESTING
I FIND YOUR LACK OF TESTS DISTURBING.

٣٦٢٤٩٣
مكتبة الابحاث

خديجان كسيور

المجمع التشخيصي للعقوبة
الطباطبات | التاريخ |
البيزاء



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

Never measure
developers by
Code Coverage!

Demo

(Coverage)

Coverage helps.

Use it!

Really, Use it!

Demo

(Jenkins)



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

Immediate Feedback and Early Warning

Some CI Servers

- Java - Jenkins (Hudson), CruiseControl
- .NET - TFS, Jenkins, CruiseControl.NET
- C++ - Jenkins (Shell/Make)
- Ruby – CruiseControl.rb
- commercial products - TeamCity, ALA

The Addison Wesley Signature Series

CONTINUOUS INTEGRATION

RELIABLE SOFTWARE
TEST, AND DELIVERY

JEZ H.
DAVIS

PAUL M. DUVALL
WITH
STEVE MATYAS
ANDREW GLOVER

The Addison Wesley Signature Series

IMPROVING SOFTWARE QUALITY
AND REDUCING RISK



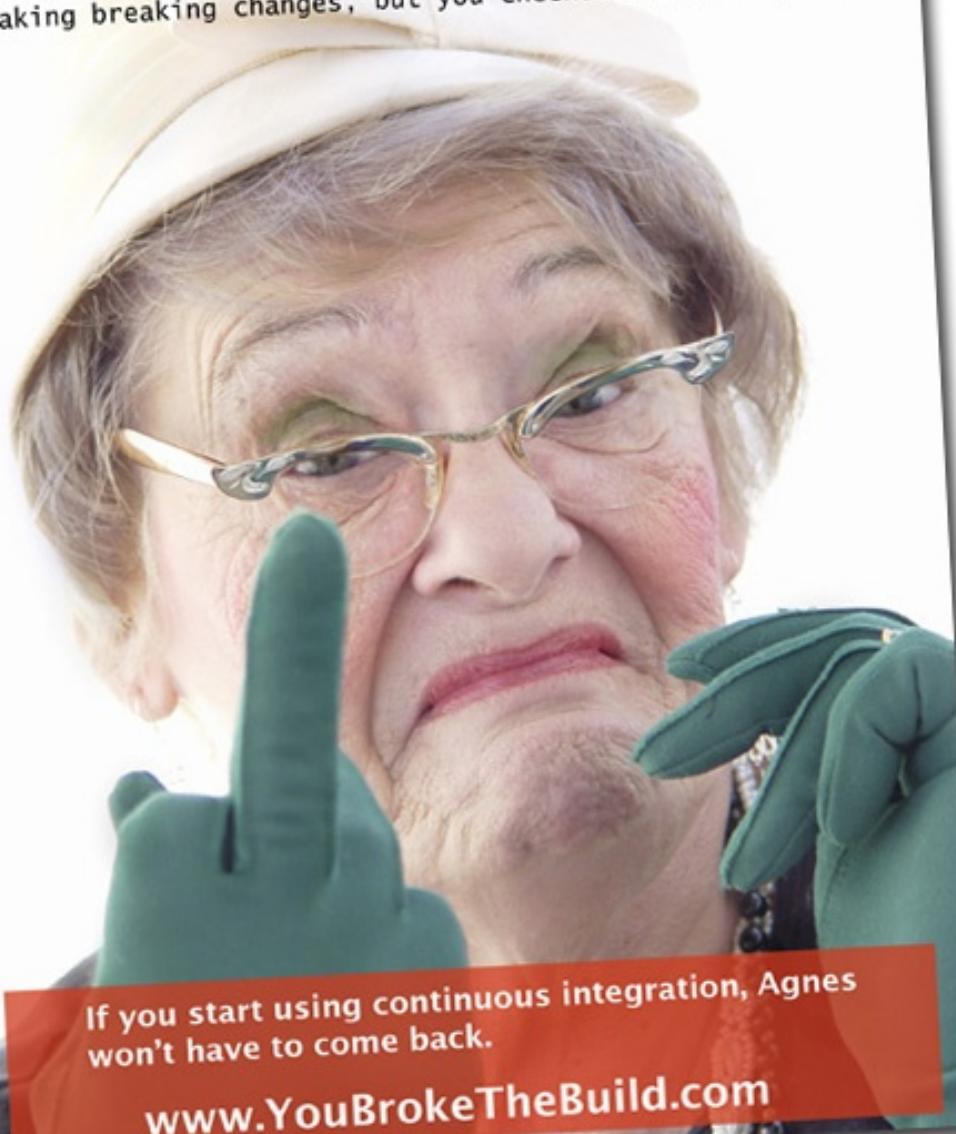
Forewords by Martin Fowler and Paul Julius

A MARTIN FOWLER SIGNATURE BOOK
Martin



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

Good Enough?

Demo

(Warnings)



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

Many, Many Tools

Good Enough?

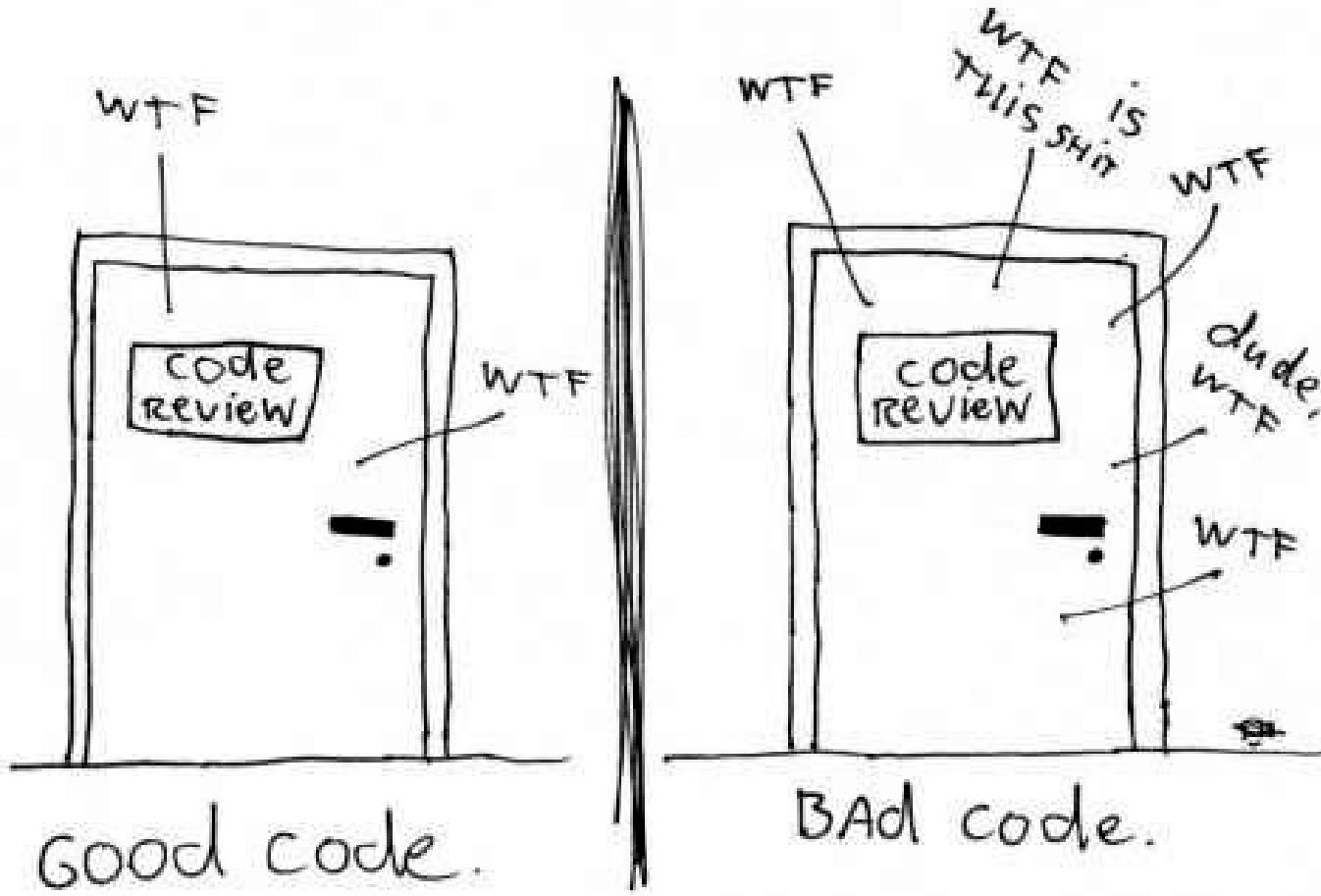
Demo

(Review)

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



Lessons Learned

(!) Boy Scout Rule

(!) Do TDD/BDD

(!) Use the Tools
(Coverage, CI, Static Analysis)

Thank
You



Peter Kofler



@codecopkofler

www.code-cop.org

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
- <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/>

Kata

- [http://en.wikipedia.org/wiki/Kata_\(programming\)](http://en.wikipedia.org/wiki/Kata_(programming))
- <https://bitbucket.org/pkofler/primefactors.java/> (Kata Source Code)

Links #2

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/>
- <http://behaviour-driven.org/>

Code Coverage

- http://en.wikipedia.org/wiki/Code_coverage
- <http://www.ibm.com/developerworks/java/library/j-cqo1316/>

Links #3

Continuous Integration

- http://en.wikipedia.org/wiki/Continuous_integration
- <http://www.codinghorror.com/blog/archives/000818.html>
- <http://www.stevemcconnell.com/ieeesoftware/bpo4.htm>
- <http://www.joelonsoftware.com/articles/fogooooooooo23.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

Code Review

- http://en.wikipedia.org/wiki/Code_review
- http://en.wikipedia.org/wiki/Pair_programming

CC Images

- spray face: <http://www.flickr.com/photos/iangallagher/4115047191/>
- Judge Dredd: <http://www.flickr.com/photos/eldave/6169431454/>
- list: <http://www.flickr.com/photos/kylesteeddesign/3724074594/>
- eggs: <http://www.flickr.com/photos/nickwheeleroz/2475011402/>
- windows: <http://www.flickr.com/photos/sketchglass/4281424410/>
- shoe maker: <http://www.flickr.com/photos/tbatty/1450209613/>
(by 2010, now dead link)
- factory: <http://www.flickr.com/photos/94693506@Noo/4643248587/>
- passion: <http://gapingvoid.com/2011/03/29/suddenly-passion/>
- questions: <http://www.flickr.com/photos/oberazzi/318947873/>

CC Images

- fence: <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/>
- wtf: <http://www.flickr.com/photos/smitty/2245445147/>
- questions: <http://www.flickr.com/photos/seandreilinger/2326448445/>