Code Quality Assurance
(summary of all presentations)

Peter Kofler, ‘Code Cop’
IBM, March 2012
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)
Boy Scout Rule
Robert C. Martin
(“Uncle Bob”)
Clean Code
A Handbook of Agile Software Craftsmanship
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
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
Test-Driven Development
By Example

Kent Beck
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
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
Continuous Integration
Improving Software Quality and Reducing Risk

Paul M. Duvall
with
Steve Matyas
Andrew Glover

Forewords by Martin Fowler and Paul Julius
Good Enough?
Demo
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, (Re)Sharper, 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 code.

Bad code.
Good Enough?
(Probably Not)
Lessons Learned

(!) Boy Scout Rule

(!) Do TDD/BDD

(!) Use the Tools

(Coverage, CI, Static Analysis)
Thank You
Links #1

Mindset
- http://pragprog.com/the-pragmatic-programmer

Software Craftsmanship
- http://clean-code-developer.de/
Links #2

Kata
- http://hg.code-cop.org/primefactors/ (Kata Source Code)

Unit Test
- http://en.wikipedia.org/wiki/Unit_testing

TDD

BDD
- http://dannorth.net/introducing-bdd/
Links #3

Code Coverage

Continuous Integration
- http://www.stevemcconnell.com/ieeesoftware/bp04.htm
- http://jenkins-ci.org/

Static Code Analysis
CC Images

- spray face: http://www.flickr.com/photos/ianagallagher/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/
- 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/
- questions: http://www.flickr.com/photos/seandreilinger/2326448445/