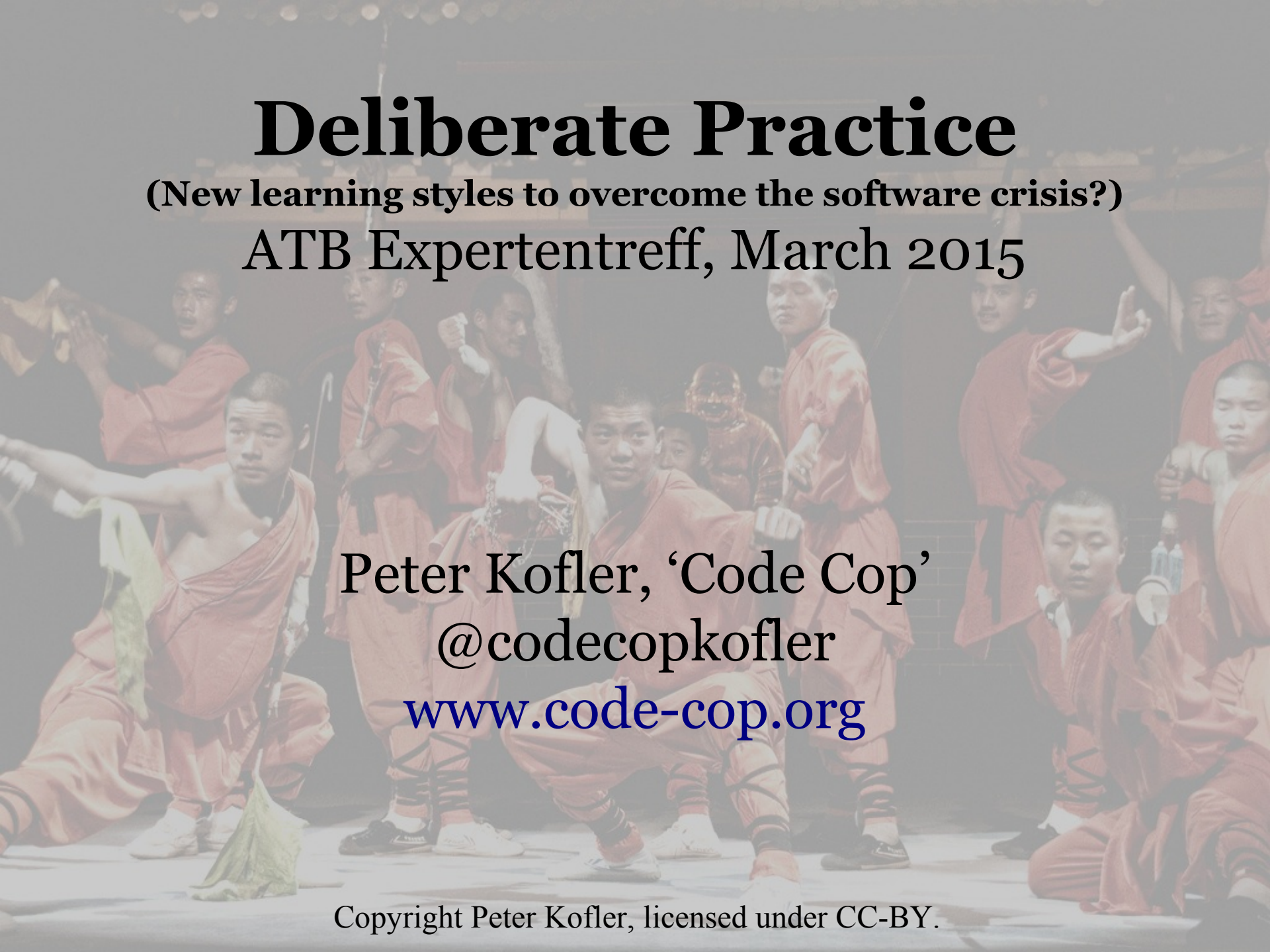


Deliberate Practice

(New learning styles to overcome the software crisis?)

ATB Expertentreff, March 2015

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

Who am I?

Peter Kofler

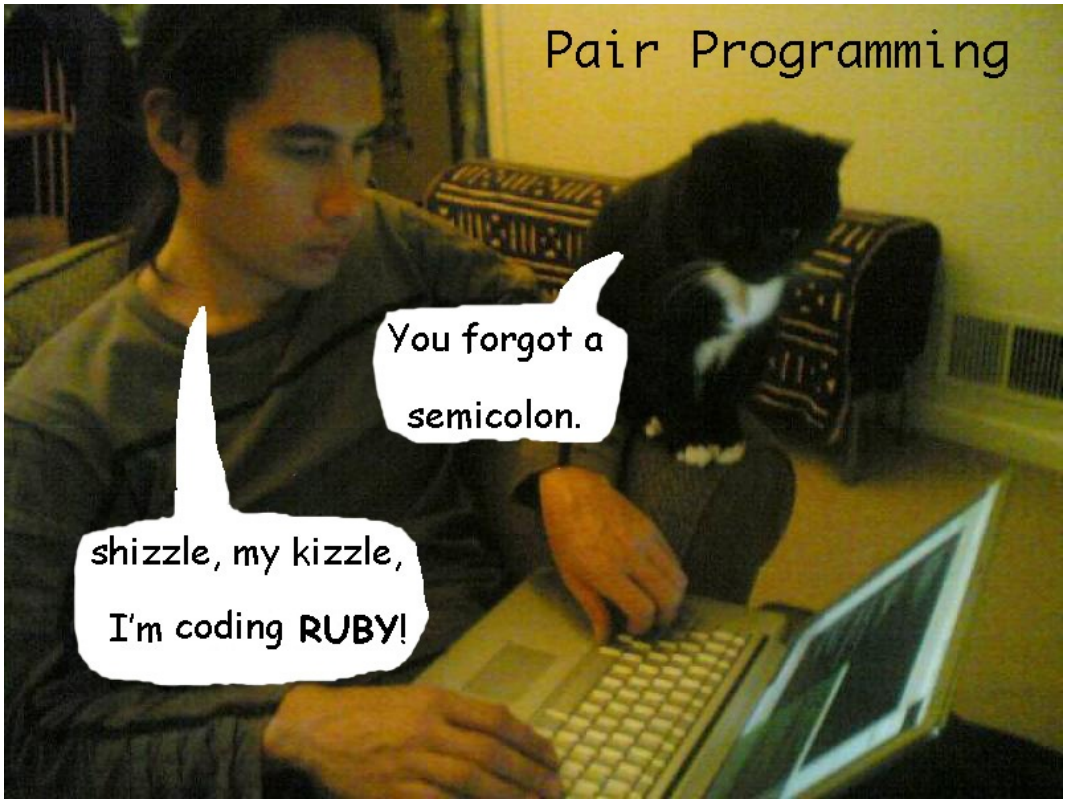
- Ph.D. (Appl. Math.)
- Professional Software Developer for 15 years
- “fanatic about code quality”

I help development teams with

- Professionalism
- Quality and Productivity
- Continuous Improvement

Training and Mentoring

- Pair Programming
- Programming Workshops
- Deliberate Practice



Who are You?

Quick Poll: Are You a

- Tester?
- Test Manager?
- QA?
- QA Manager?
- Developer?
- Architect?

No, you are a Software Delivery Professional!

Your Goal: Developing Quality Software

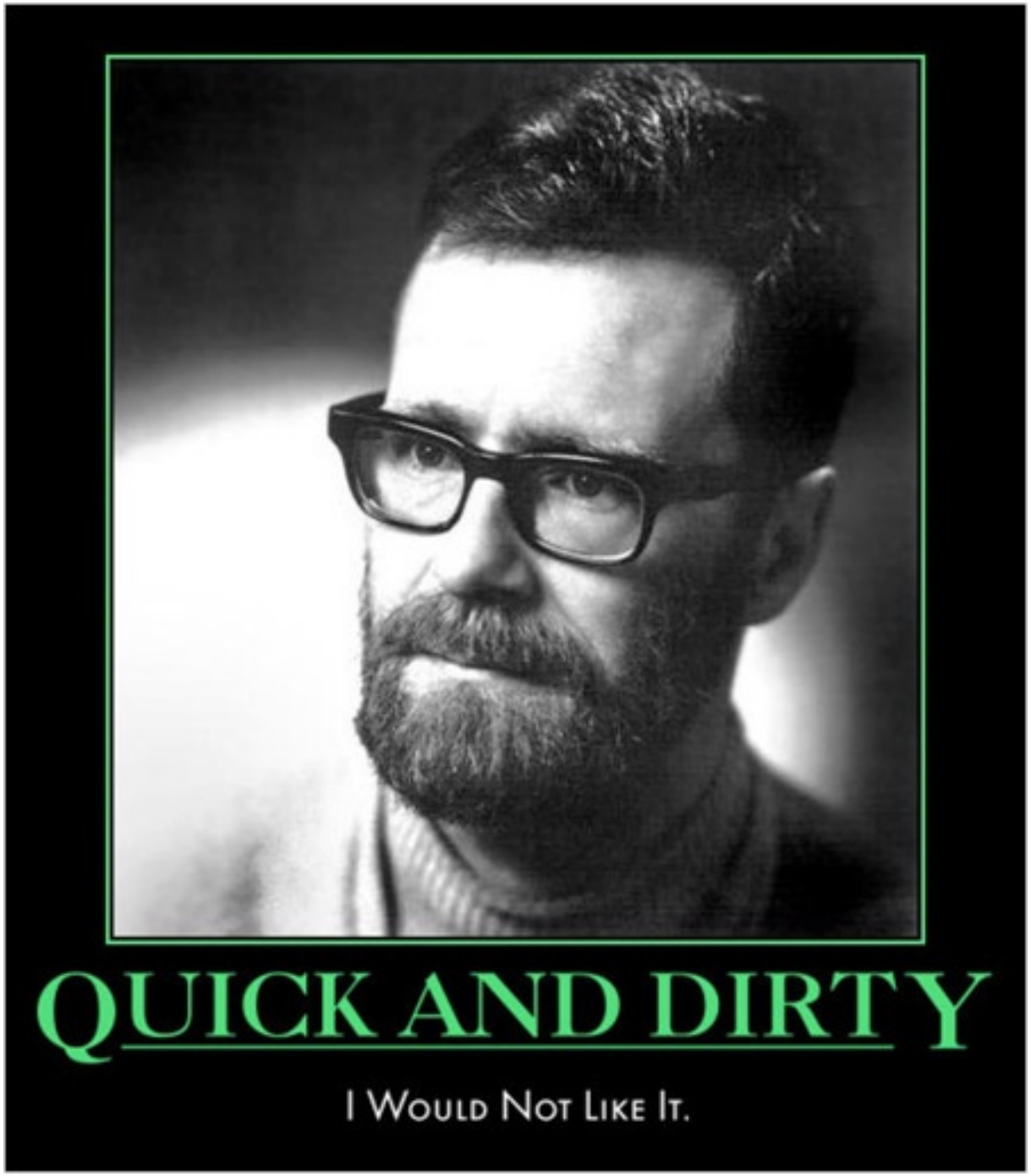

So What Is the Problem?

Software Crisis?

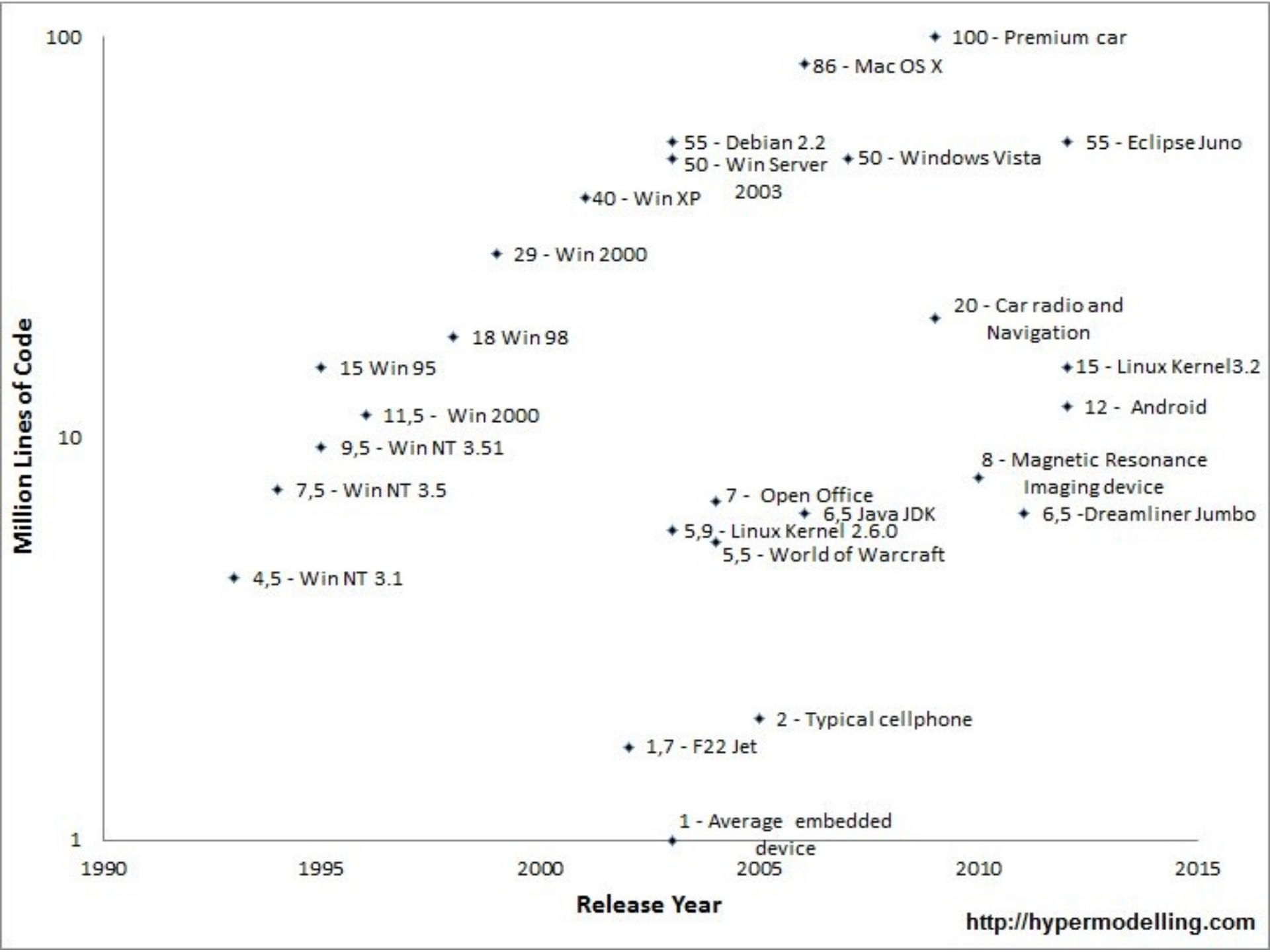
Software Crisis

"The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem."

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


QUICK AND DIRTY
I Would Not Like It.

Requirements and Complexity Increase Each Year


100
10
1
Million Lines of Code
100 - Premium car
86 - Mac OS X
55 - Debian 2.2
50 - Win Server 2003
50 - Windows Vista
55 - Eclipse Juno
40 - Win XP
29 - Win 2000
20 - Car radio and Navigation
18 Win 98
15 Win 95
15 - Linux Kernel3.2
11,5 - Win 2000
12 - Android
9,5 - Win NT 3.51
8 - Magnetic Resonance Imaging device
7,5 - Win NT 3.5
7 - Open Office
6,5 Java JDK
6,5 -Dreamliner Jumbo
5,9 - Linux Kernel 2.6.0
5,5 - World of Warcraft
4,5 - Win NT 3.1
2 - Typical cellphone
1,7 - F22 Jet
1 - Average embedded device
1990
1995
2000
2005
2010
2015
Release Year
http://hypermodelling.com

New Technologies Move Very Fast

Knowledge Half-Life of 18 Months

http://www.oracle.com/technetwork/articles/javase/kabutz-qa-136652.html

But Abstractions are Leaky

- What is HTTP?
- What is a pointer?
- What is Assembler?
- What is RISC/ CISC?

Enormous Legacy

- e.g. still 200 billion LoC COBOL (2008)
- "Java is the new COBOL"

http://skeptics.stackexchange.com/questions/5114/did-cobol-have-250-billion-lines-of-code-and-1-million-programmers-as-late-as-2


We are crushed

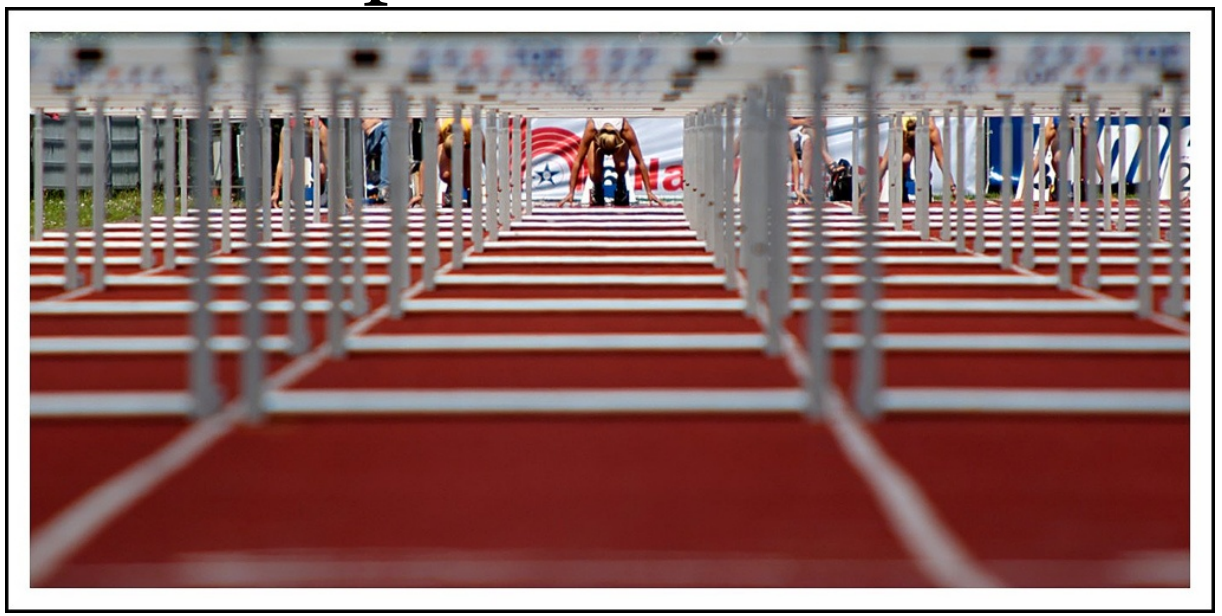
What can be done about it? How to overcome the „Crisis“?

More Metrics?

- Lines of Code?
- Hours worked?
- Velocity?
- Code Coverage?
- Defects / time?
- …

More Process/Methodology?

- Software Engineering
- Extreme Programming
- Agile Software Development
- Scrum
- Kanban
- What's next?

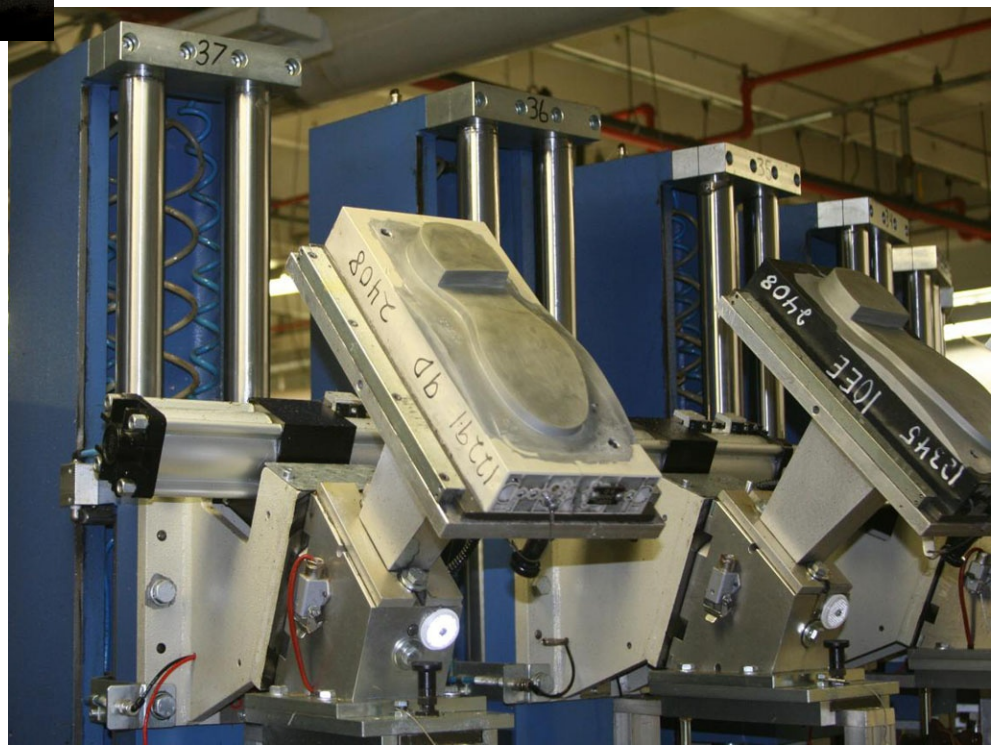



MOAR TESTS!!1!
MOAR!!1!
memegenerator.net

Who (in the end) creates software?

Professional
(people)

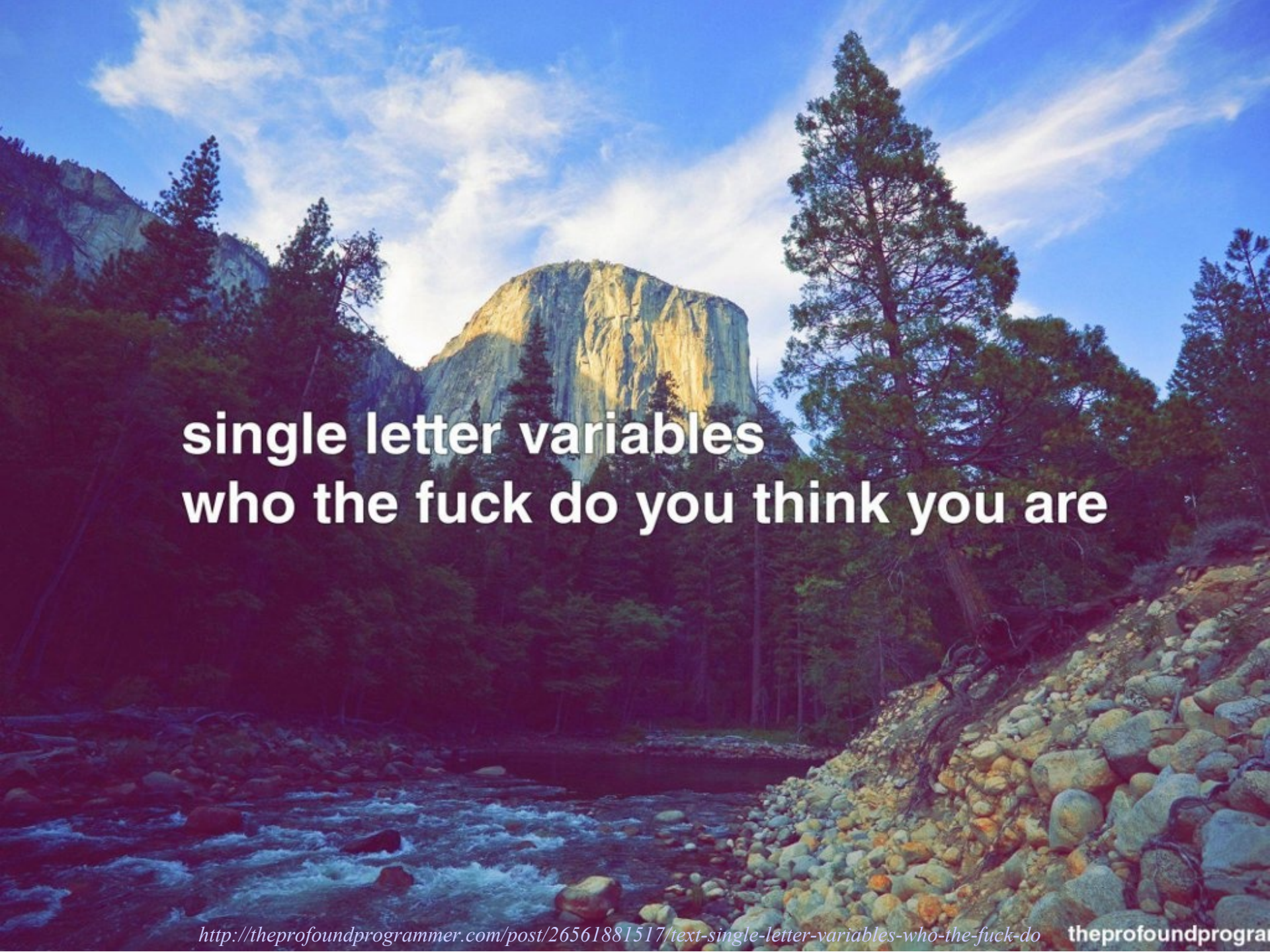
Engineering
(process)



Software Professionals Create Software!

Software like that

```
@SuppressWarnings({
  "UnusedDeclaration",
  "AssignmentToDateFieldFromParameter",
  "AssignmentToCollectionOrArrayFieldFromParameter",
  "ClassWithTooManyMethods",
  "ClassWithTooManyFields",
  "OverlyComplexClass"})
public class OewDocument { // NOPMD
    …
}
```


single letter variables
who the fuck do you think you are
http://theprofoundprogrammer.com/post/26561881517/text-single-letter-variables-who-the-fuck-do
theprofoundprogra

or like that

```
...
} catch (Throwable t) {
    throw new Error(t.getMessage());
}
...
for (Throwable e : exceptionList) {
    throw e;
}
```

Why is its quality so bad?

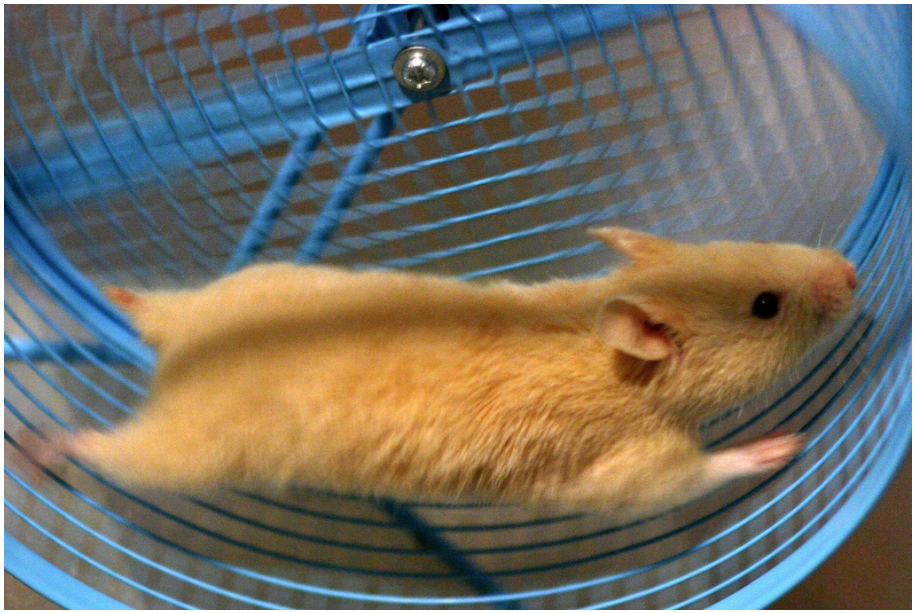
Is Software Engineering Education sufficient?

http://programmers.stackexchange.com/questions/60595/did-your-cs-program-prepare-you-well

Training on the Job?

Yes, some but...

- only what is already there
- Trial & Error not popular in production
- no practice - only production
- time pressure



Our Industry is Very Young

- Half of all developers age < 30

http://blog.cleancoder.com/uncle-bob/2014/06/20/MyLawn.html

Not Enough Senior People

- either busy
    - e.g. critical issues
    - contact with business
- or left active development
    - architects,
    - analysts,
    - managers etc.

Developing Quality Software Developers

Quick Poll: Do you ...

- Technical magazines?
- Internal library?
- Reading groups?
- Lunch & Learn?
- Conferences?
- Trainings?
- Hackdays?

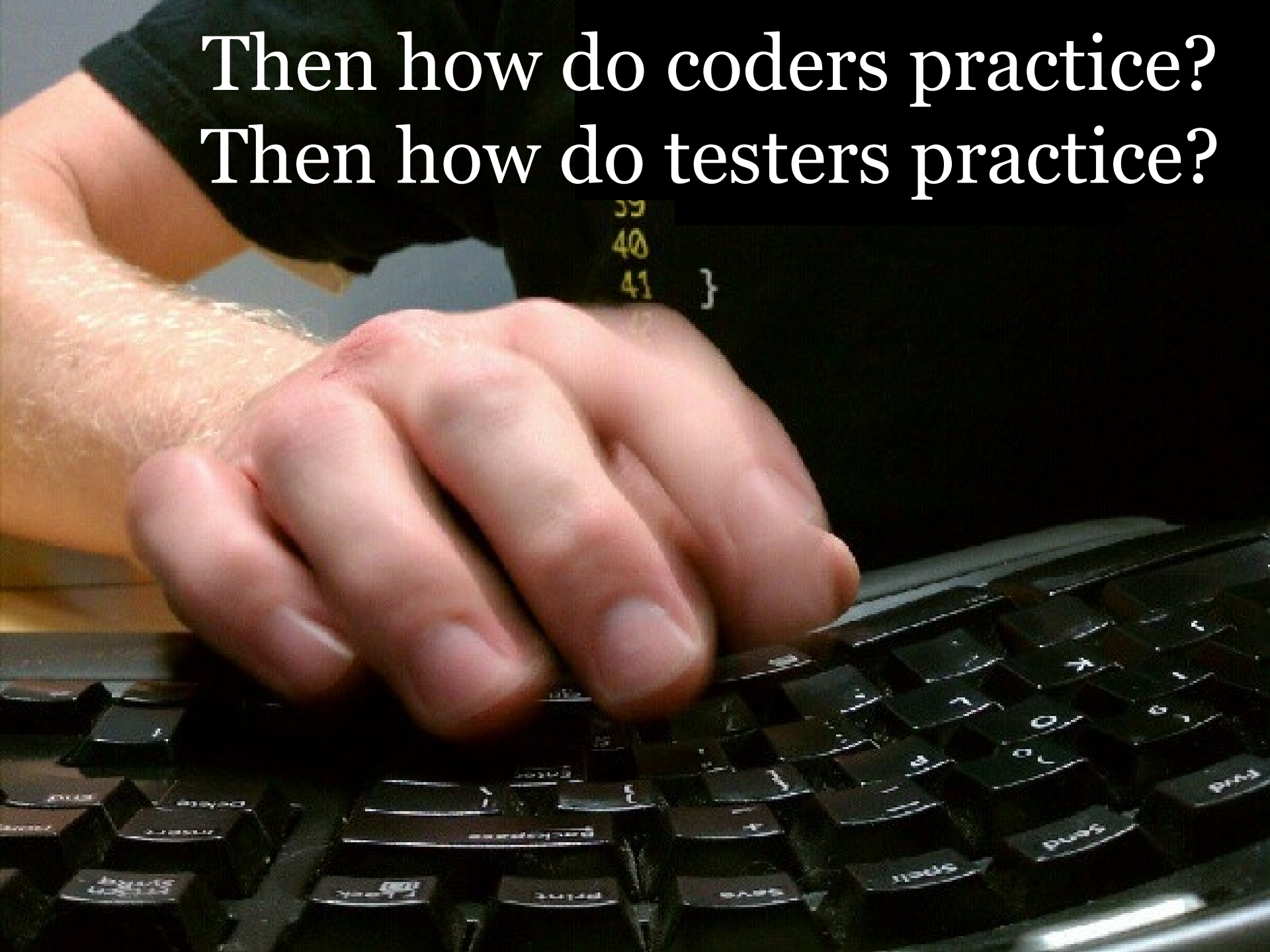
Classic Training is “Sheep Dip“

That is not enough


MOAR PRACTICE!!1!
MOAR!!1!
memegenerator.net

Deliberate Practice

How do musicians practice?


Then how do coders practice?
Then how do testers practice?


Code Kata

Code Kata Definition

- A kata as a detailed choreographed pattern of movement.
- A code kata is an exercise in programming which helps a programmer hone their skills through practice and repetition.
- A testing kata is ...

Code Kata

- simple problem (max. 20 minutes)
- solve every day
    - memorization/create reflexes
- experiment with solutions
    - stretch yourself
    - try new technologies

Why repeat the same kata?

- de-emphasise the generation of code
- concentrate on
    - the process of writing the code
    - naming test cases
    - the Red/Green cycle

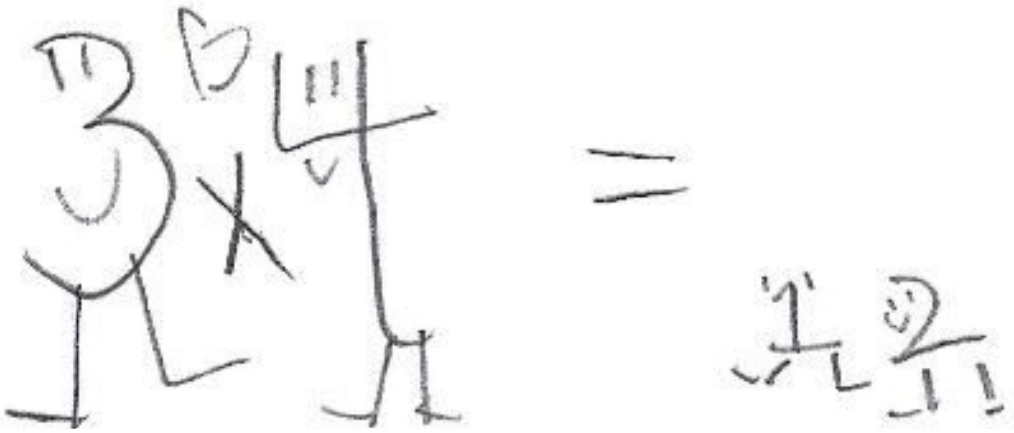
Don't Focus on Getting it Done. Focus on Doing It Perfectly.

Software Katas

- Finding algorithms
- Coding solutions
- Bringing code under test
    - Unit Test/Test tools (“Test Lab“)
- Refactoring katas
- Finding test cases
- Architectural katas

Code Kata Example: Prime Factors

Draw a picture to illustrate the multiplication $3 \times 4 = 12$.



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

Prime Factors Statistics

- very easy
  - 10 minutes
  - 6 test cases
  - final algorithm is 5 lines
- I did it 100+ times

To learn and practice

- Test Driven Development cycle
- first in every new language
- IDE short-cuts
- laptop keyboard layout
- focus when distracted

Kata Example: Car Mechanic

What else could we practice like that?


Coding Dojo

Coding Dojo Definition

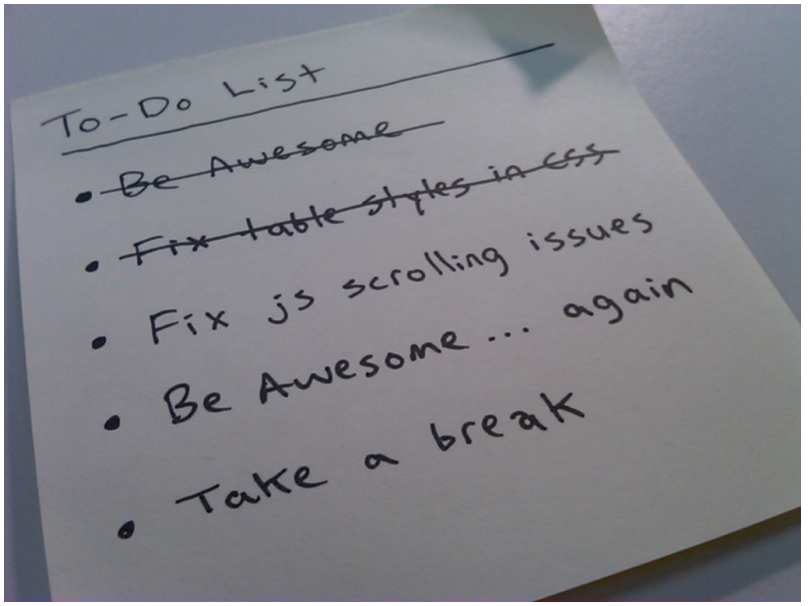
- A coding dojo is a meeting where a bunch of coders get together and work on a code kata, a programming challenge to improve their skills.

- They code, learn and have fun away from interruptions, distractions, deadlines and production bugs.

http://codingdojo.org/

Coding Dojo Structure

- Introduction 15'
- Coding 45'
- Interim (Retrospective) 5'
- Break 15'
- Coding 45'
- Retrospective 15'



Coding Dojo Mindset

- Safe place outside work
- We are here to learn
- Need to slow down
- Focus on doing it right
- Collaborative Game

Coding Dojo Rules

Pair Programming

- Collaborative = Pair Programming
  - “Randori“ (pairing on the projector)
  - or regular programming in pairs

- regular Pair Programming
  - do not talk for too long
  - do not interrupt the other
  - no “keyboard hugging“

Pair Programming adds discussion & a second opinion to the practice.

Constraints

- Challenges during a dojo or code retreat.
- Moving to the extreme is a way of learning
- Examples
    - Missing Tool (No Mouse, …)
    - Missing Feature (No IFs, …)

Testing Dojo Example: Testcases for Gilded Rose

Given

- Production code (60 lines)
- Detailed requirements (1 page A4)
- No tests (at least 38 cases)
- Prepared FitNesse fixtures (Java)
- dbFit connector (PL/SQL)

https://bitbucket.org/pkofler/gilded-rose-fitnesse

Testing Task

- Modify existing test cases, play around
- Create some “perfect” tests
  - derive test cases from requirements
  - name test cases accordingly
  - test boundary conditions
  - readable, concise, free of duplication
- Experiment with styles

Dojo Example: Data Dojo

- Study group / workshops
- Learning and sharing knowledge
- Big data technologies.
- Currently focused on Apache Spark.

http://www.meetup.com/datadojo/

Coding Dojo Vienna

- Monthly, free Coding Dojo in Vienna
- Weekdays after work
- Changing locations
- Follow #CodingDojoVie

http://codingdojovie.github.io/

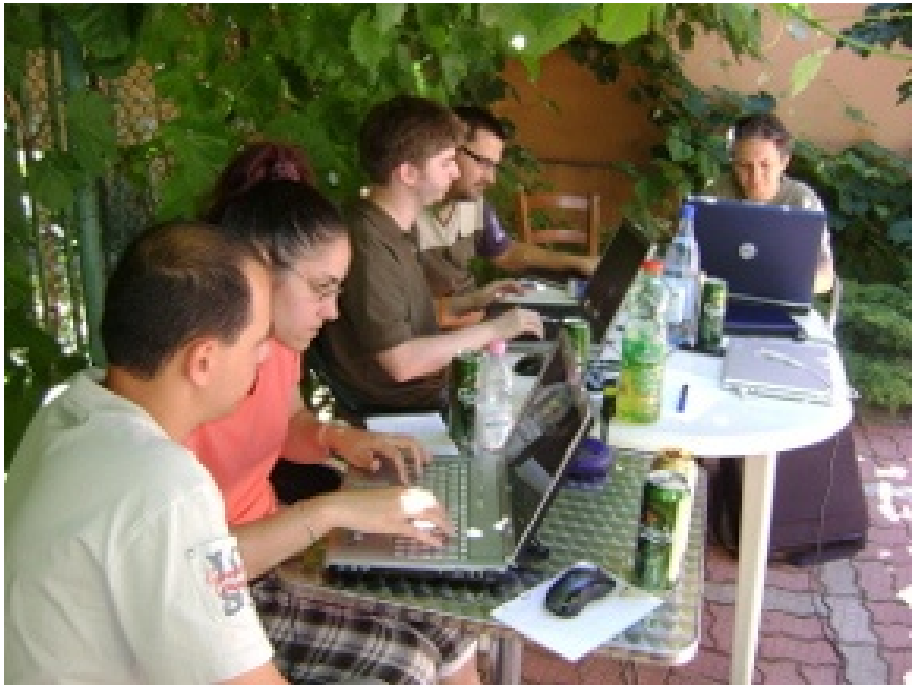

Code Retreat

Code Retreat

- A day-long, intensive practice event, focusing on the fundamentals of software development and design.
- Practising the basic principles of modular and object-oriented design.
- Support mindset of quality, learning and practice!

How it started

- CodeMash conference 2009
- Popularised by Corey Haines
- 2009 also first time outside US (Romania)



Code Retreat Structure

- Introduction 15'
- 3 Sessions
  - Coding 45'
  - Retrospective/Break 15'
- Lunch 60'/90'
- 3 Sessions
- Retrospective 45'

Code Retreat Principles

- Language agnostic
- Learn through pairing (switch pairs)
- Delete code after each session
- Practice
- Experiment
- Have fun!

Coderetreat

- honing the craft together

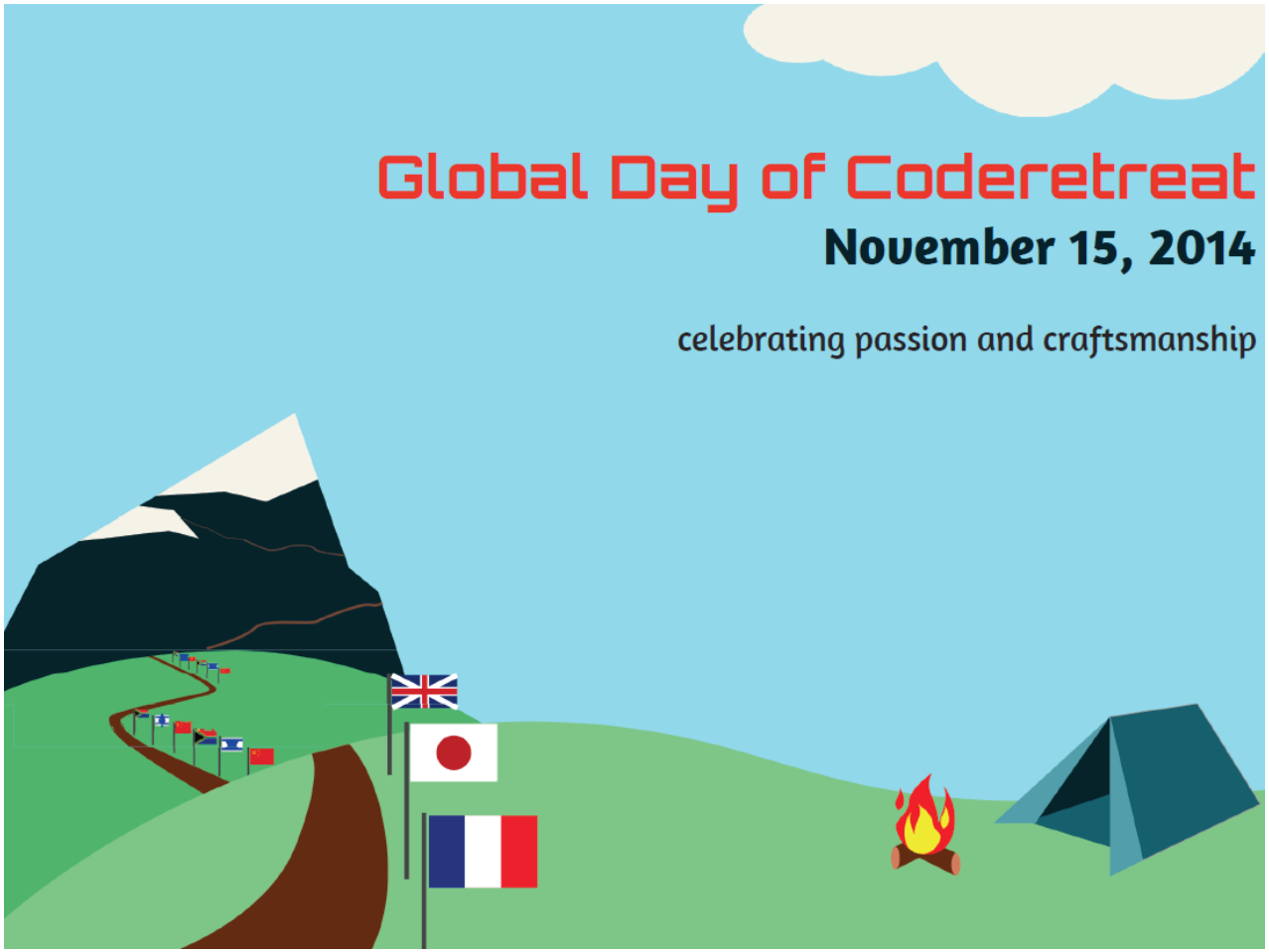
http://coderetreat.org/

Why delete the code?

- No. Listen.
- Stop trying to go faster, start trying to go slower.
- Don't think about finishing, think about improving.
- Think about practising. As a team.
- That's what this day is for. Nothing else.

http://www.cyber-dojo.com/

Code Retreat Example: GDCR



Global Day of Code Retreat

- A world-wide event celebrating passion and software craftsmanship.
- Once a year, November/December
- Whole Saturday, starting 9:00
- Look for #GDCR15

http://globalday.coderetreat.org/

Example: Test Automation Retreat

Conclusion

Developing Quality Software

Software Professionals Create Software!

Developing Quality Software Developers

Deliberate Practice


CALL TO
ACTION

What we “Experts” need to do

- Make space for deliberate learning
- 1st Improve yourself
  - Try katas
  - Visit dojos/retreats
- 2nd Mentor next generations
  - Prepare kata exercises
  - Run in-house dojos/retreats

We will not ship shit!

(Uncle Bob)

Peter Kofler

@codecopkofler

www.code-cop.org

CC Images #1

- masters https://www.flickr.com/photos/kevinpoh/4404925960
- Bruce http://www.flickr.com/photos/sherpas428/4350620602/
- pairing http://www.flickr.com/photos/dav/94735395/
- wants you http://www.flickr.com/photos/shutter/105497713/
- Cthulhu http://community.imaginefx.com/forums/thread/402575.aspx
- leaky wall https://www.flickr.com/photos/gammaman/7803857922
- cars https://www.flickr.com/photos/24354425@N03/14575356310
- tank https://www.flickr.com/photos/mangee/141936439
- dump http://www.flickr.com/photos/sanmartin/2682745838/
- measures https://www.flickr.com/photos/bigtallguy/153619774
- hurdles https://www.flickr.com/photos/robert_voors/774926895

CC Images #2

- shoe maker http://www.flickr.com/photos/tbatty/1450209613/ (by 2010)
- factory http://www.flickr.com/photos/94693506@N00/4643248587/
- Hamster http://www.flickr.com/photos/zebrapares/4529836138
- baby http://www.flickr.com/photos/11904001@N00/3983980813/
- sheep dip https://www.flickr.com/photos/christianspenceranderson/6990570682
- music https://www.flickr.com/photos/jeremyhiebert/11474303315
- code https://www.flickr.com/photos/slworking/7726767220/
- karate http://www.flickr.com/photos/the-consortium/5301698212/
- math https://www.flickr.com/photos/jimmiehomeschoolmom/4427775569
- mechanic https://www.flickr.com/photos/ravages/9242464882
- dojo http://www.flickr.com/photos/49715404@N00/3267627038/

CC Images #3

- todos http://www.flickr.com/photos/kylesteeddesign/3724074594/
- rule http://www.flickr.com/photos/phunk/4188827473
- inn http://www.flickr.com/photos/danielleblue/170496395/
- incense http://www.flickr.com/photos/cibomahto/7651263514/
- automation http://www.flickr.com/photos/aquilaonline/510921786/
- finish http://www.flickr.com/photos/jayneandd/4450623309/
- call https://www.flickr.com/photos/melenita/15166203826