

# Deliberate Practice

(New learning styles to overcome the software crisis?)

Agile Slovenia 2015

Peter Kofler, 'Code Cop'  
@codecopkofler  
[www.code-cop.org](http://www.code-cop.org)

Copyright Peter Kofler, licensed under CC-BY.

# Peter Kofler

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



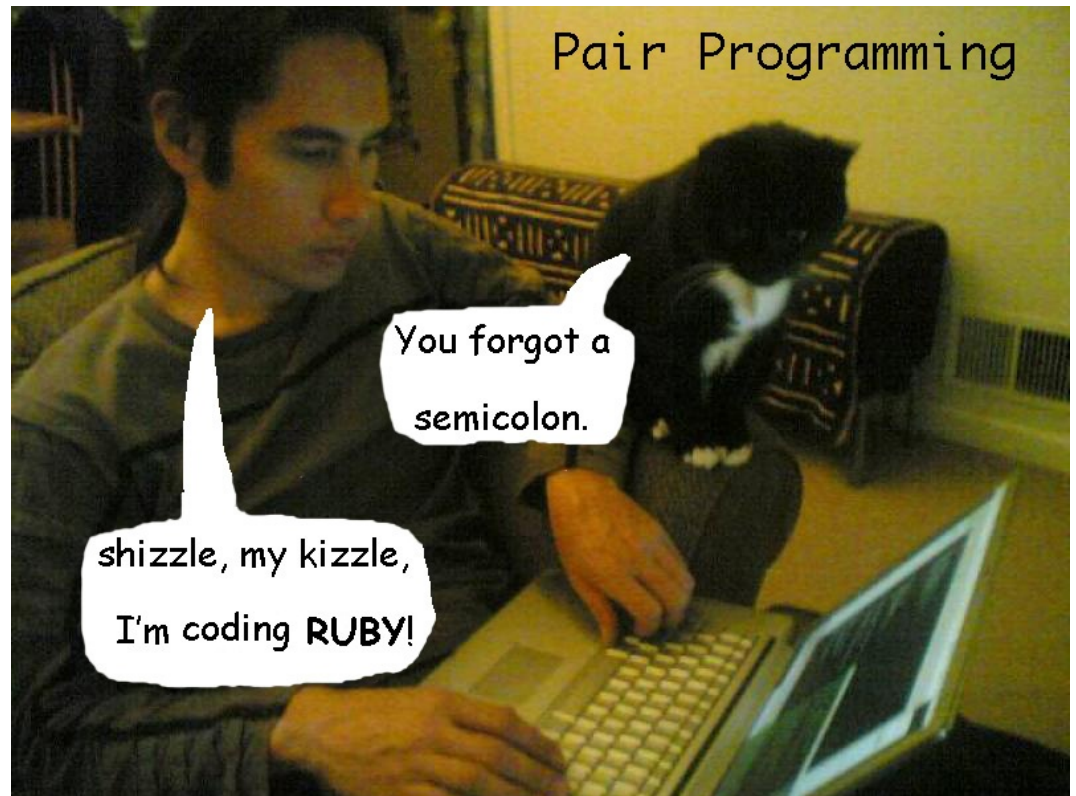
# I help development teams with

- Professionalism
- Quality and Productivity
- Continuous Improvement



# Training and Mentoring

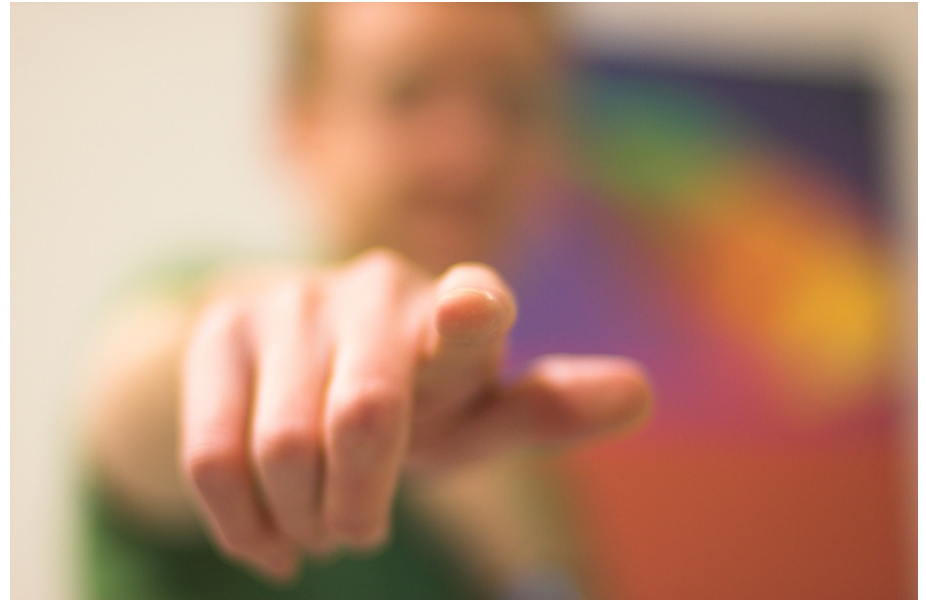
- Pair Programming
- Programming Workshops
- Deliberate Practice





# Who are You?

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



No, you are a  
Software Delivery  
Professional!

Your Goal:  
Developing  
Quality  
Software



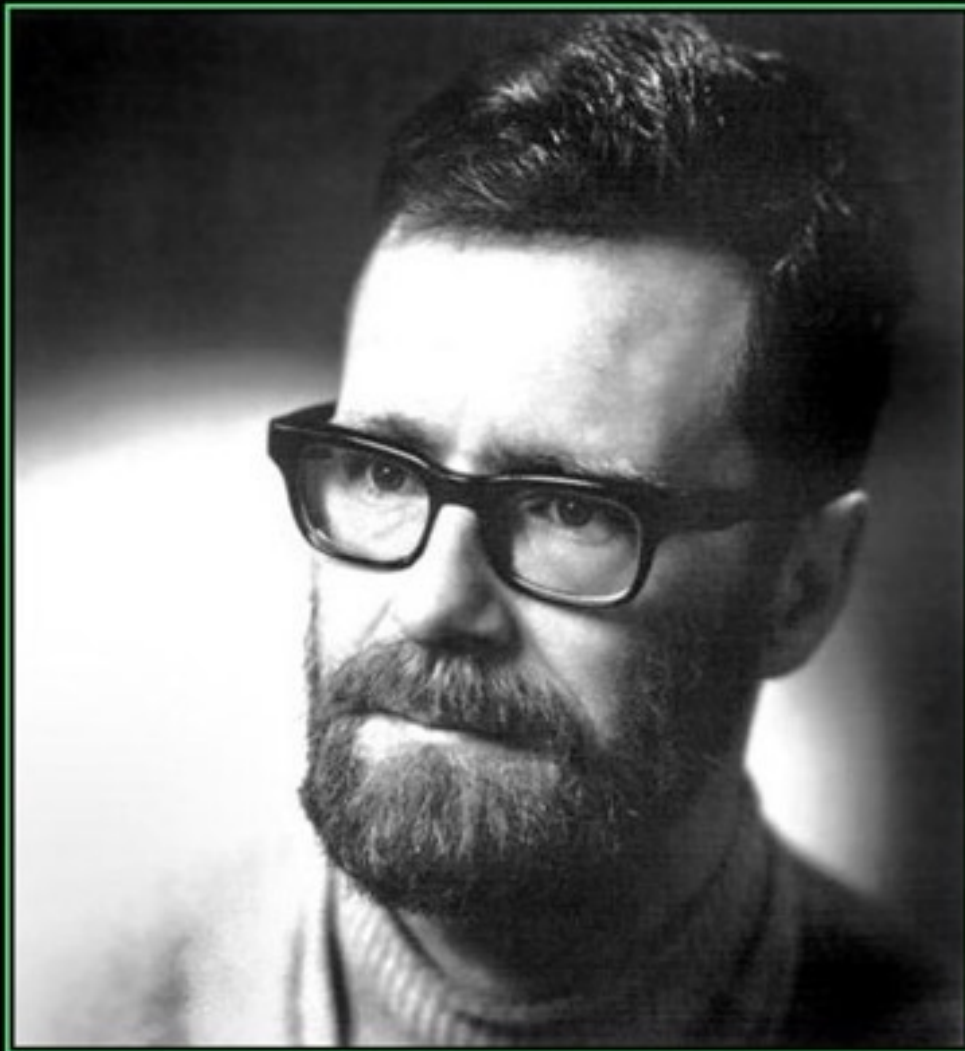
What's the Problem?



# 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.”



# QUICK AND DIRTY

I WOULD NOT LIKE IT.

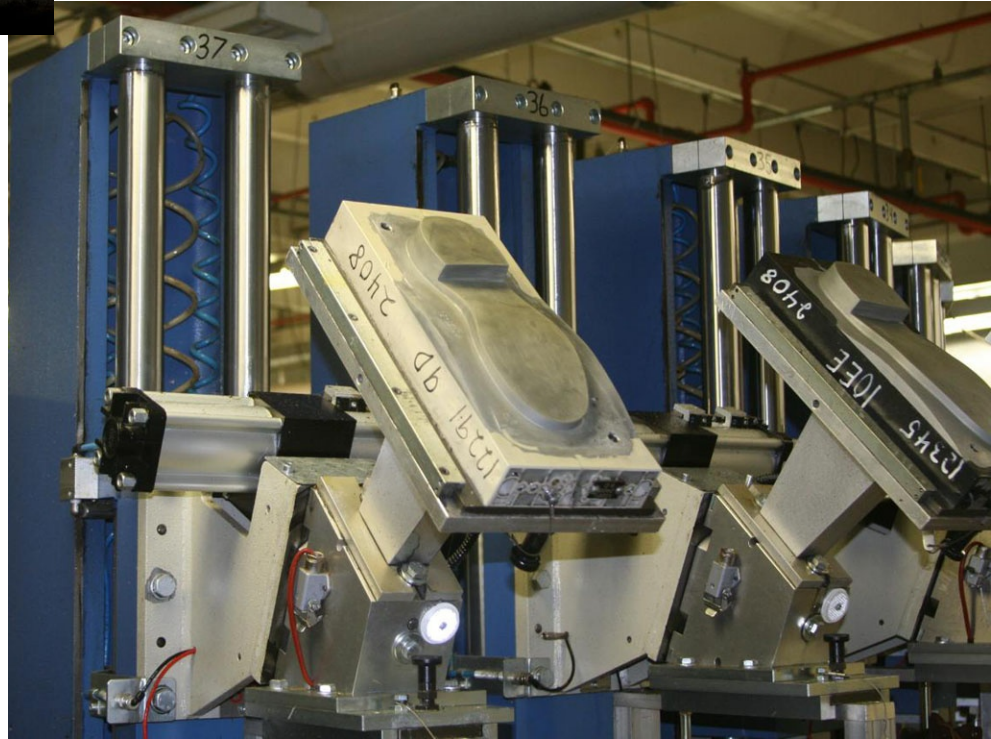
Who creates  
this software  
he talked about?





Professional  
(people)

Engineering  
(process)



Software  
Professionals  
Create Software!

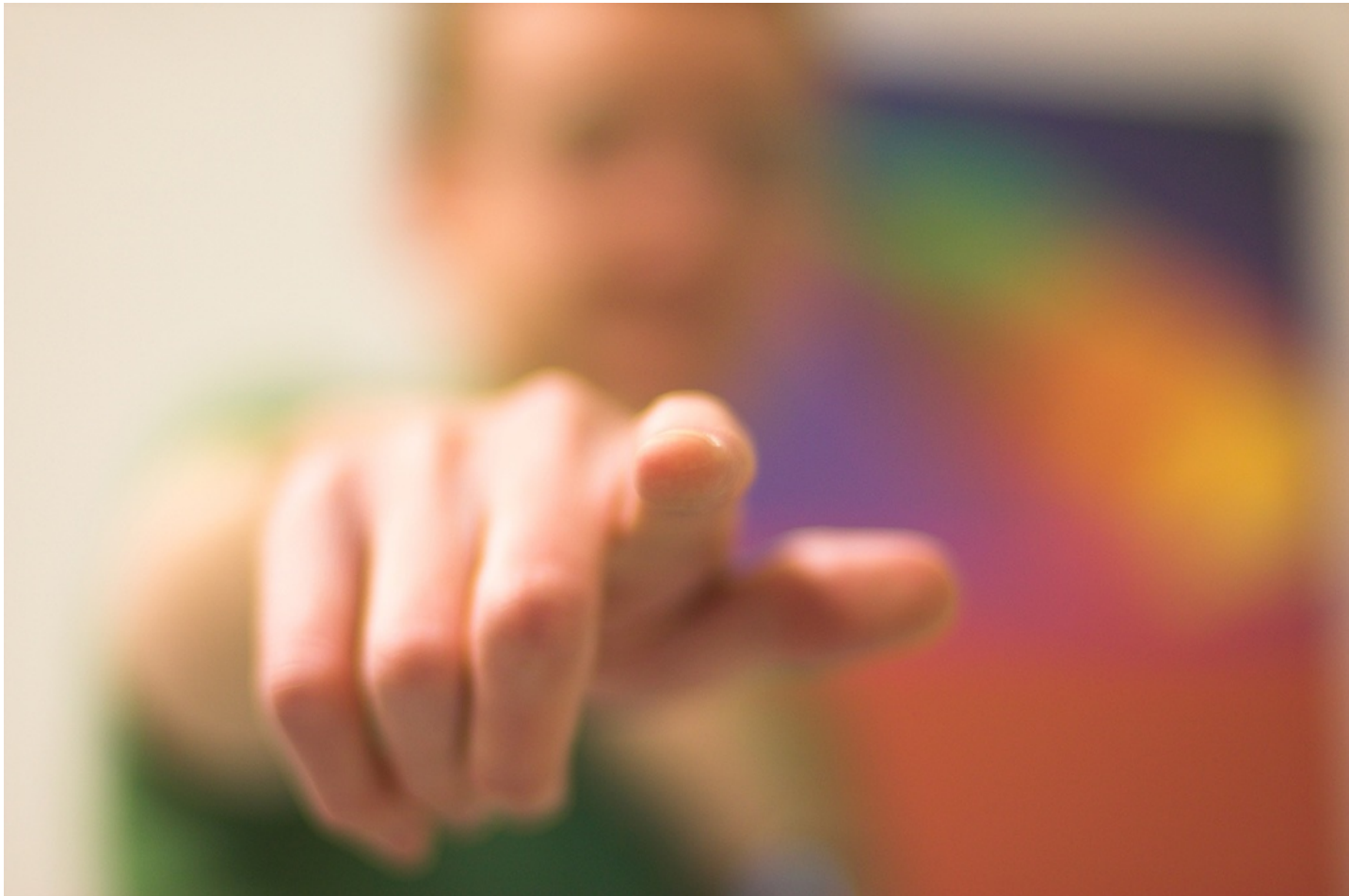


Software like that...





# Why is its quality so bad?





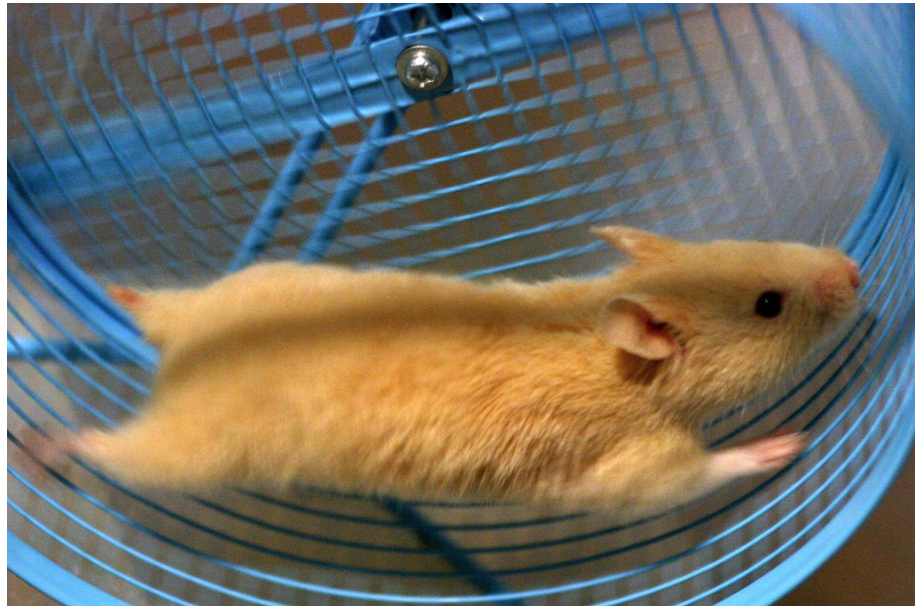
Is Software  
Engineering  
Education  
sufficient?

# 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





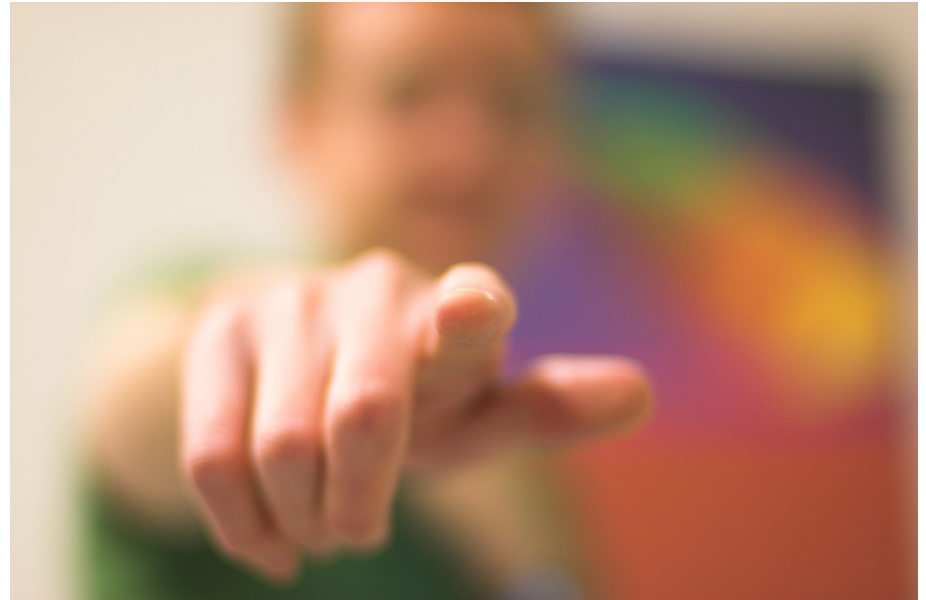
# Not Enough Senior People

- 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”



DIPPING SHEEP.

*Dallas, Texas. Mar. 6. 1907*

That is not enough

**MOAR PRACTICE!!1!**

**MOAR!!1!**



# 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

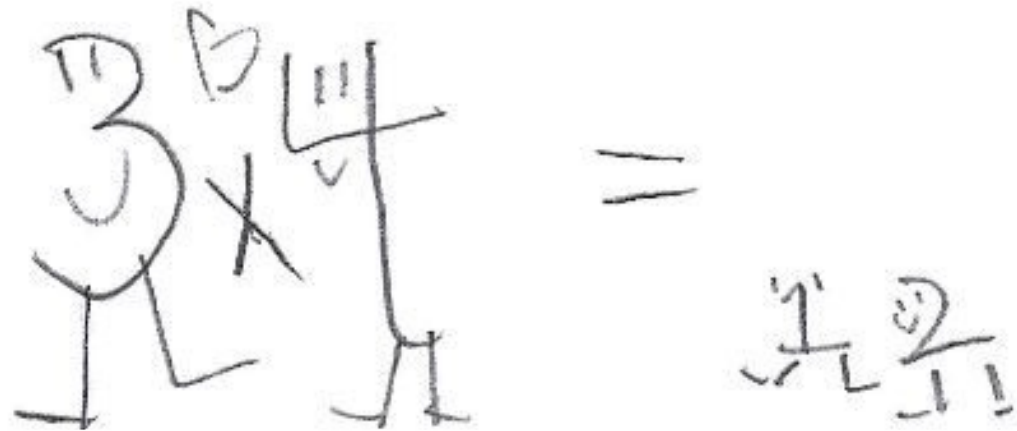
Don't Focus on  
Getting it Done.  
Focus on Doing  
It Perfectly.

# 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.

# I did it 115 times

- very easy
  - 10 minutes
  - 6 test cases
  - final algorithm is 5 lines

# 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





# Coding Dojo

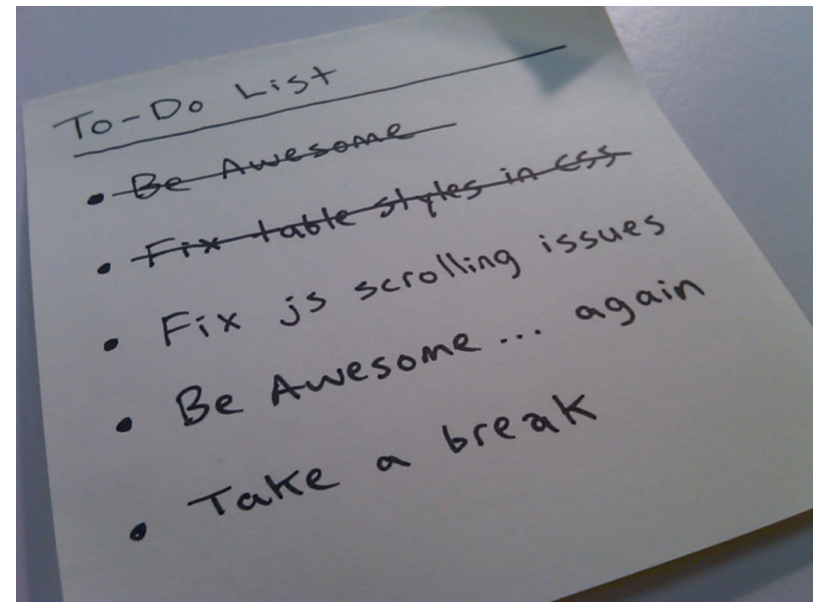


# Coding Dojo Definition

- 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.

# 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)

# 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



# Code Retreat

# Code Retreat (Overview)

- Sort of six Coding Dojos in sequence.
- A day-long, intensive practice event.
- Examples:
  - Global Day of Coderetreat
  - Test Automation Retreat

# Conclusion





# Developing Quality Software

Software  
Professionals  
Create Software!

# Developing Quality Software Developers

# Deliberate Practice

**CALL TO  
ACTION**





# What **you** need to do

- Make space for deliberate learning
- Improve yourself
  - Try katas
  - Visit dojos/retreats
- Mentor next generation
  - Prepare kata exercises
  - Run in-house dojos/retreats

We will not  
ship shit!

(Uncle Bob)



# Peter Kofler



# @codecopkofler

# [www.code-cop.org](http://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>
- dump <http://www.flickr.com/photos/sanmartin/2682745838/>
- shoe maker <http://www.flickr.com/photos/tbatty/1450209613/> (by 2010)
- factory <http://www.flickr.com/photos/94693506@Noo/4643248587/>
- Hamster <http://www.flickr.com/photos/zebrapares/4529836138>
- baby <http://www.flickr.com/photos/11904001@Noo/3983980813/>
- sheep dip <https://www.flickr.com/photos/christianspenceranderson/6990570682>
- music <https://www.flickr.com/photos/jeremyhiebert/11474303315>

# CC Images #2

- 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/>
- 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/>
- finish <http://www.flickr.com/photos/jayneandd/4450623309/>
- call <https://www.flickr.com/photos/melenita/15166203826>