

# Pair Programming

Ruby User Group Linz, May 2015



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

# Peter Kofler

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



# Training on the Job?



# Yes, some but...

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



# I help development teams with

- Professionalism
- Quality and Productivity
- Continuous Improvement





# Mentoring

- Pair Programming
- Programming Workshops
- Deliberate Practice, e.g. Coding Dojos, Code Retreats
- Code Reviews, Architecture Discussions

# Pair Programming

A man with dark hair is sitting at a desk, focused on his laptop. A black and white cat is perched on his left shoulder, looking towards the camera. The scene is dimly lit, with the primary light source being the laptop screen. Two speech bubbles are overlaid on the image, one from the man and one from the cat.

You forgot a  
semicolon.

shizzle, my kizzle,  
I'm coding **RUBY!**

# Pair Programming

- **Two** programmers
- work **together**
- on the **same** thing
- at **one** workstation.

Pair programming is a **dialogue** between two people simultaneously programming (and analysing and designing and testing)



Enterprise CTO  
not impressed by  
pair programming.

“We often have hundreds  
of developers working  
on the same thing.”

(Hacker News Onion)

# What are the benefits?



# Benefits

- most immediate feedback
- maintained self-discipline
  - we keep each other honest
- improved code quality, less defects
- knowledge sharing
  - higher team bus number
  - mentoring junior team members.

“One disadvantage of  
pair programming: you  
have to go the toilet in  
pairs too :-)”

(Zsolt Fabók)

# Pairing Roles





# Driver

- Writes code
- Takes the decisions
- Consults with the Navigator
- Focuses on short term actions
- Verbalizes all the performed actions
- Less-informed pair member is driving

# Navigator

- Looks ahead for strategy
- Rechecks facts of the Driver (review)
- Remembers facts to the driver
- Asks meaningful questions
- Verbalizes agreement or disagreement
- Supports driver's decisions
- Does not cross the Driver's comfort zone

Students ...

★ must not listen to a music player in class.

★ must remember our teacher's name

★ must not play games in class.

★ should try to raise our hands in class.

★ should always sit in the front of the class.

mary

The Rules

# You Pair For...

- Writing Production Code (XP Rule)
- Any development on the mainline of delivery (writing code, writing tests, build engineering, documentation)
- Works well for web applications

# Pairing is Useful For...

- Tasks that need focused energy, brainstorming, problem solving, continuous design, mentoring and motivation.
- Pair by default and work solo by exception. (Novice Rule)



# When Not to Pair

- Non-mainline tasks like spiking (prototyping), research, admin.
- Tasks which call for quiet, steady focus.
- Don't insist on pairing for every task.

# How Long

- 70% of time, not more than 5 or 6 hours
  - Breaks!
- Shift partners often
  - every hour to every couple of hours
- Pair Marriage is bad
- Need quiet work time.
  - When interrupted, the negative impact is doubled.

# Workstation

- Partners sit comfortably side-by-side.
- Level 0: Basic Pairing
- Level 1: Good Pairing - two keyboards
- Level 2: Great Pairing - mirrored screen
- Level 3: Awesome Pairing - eye contact
- Level 4: “Pair”-adise - extra laptop

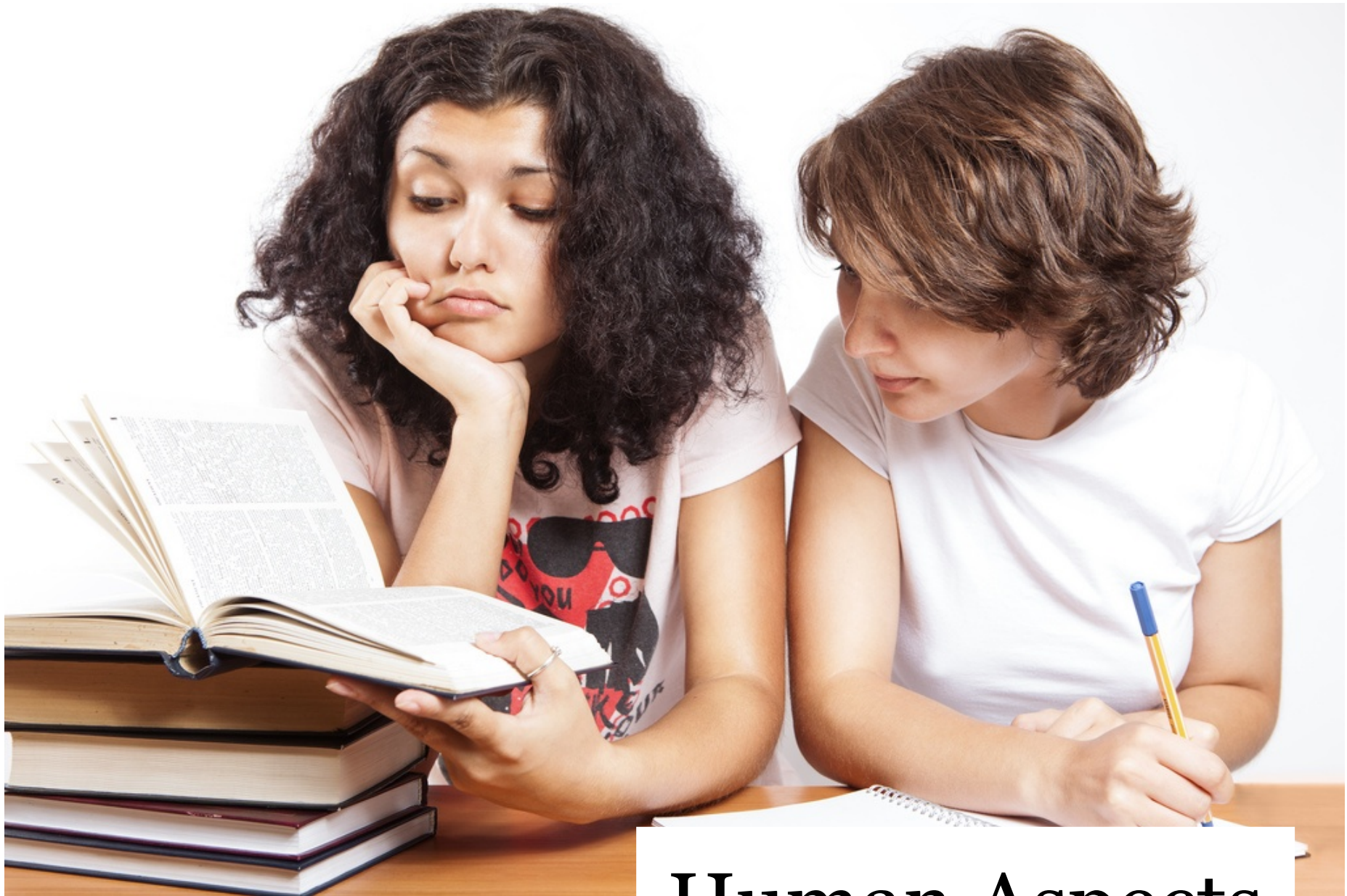
# Mob Programming

- All the brilliant people
- working at the **same** time,
- in the **same** space,
- at the **same** computer,
- on the **same** thing.



Illustration © 2012 - Andrea Zuill

<http://mobprogramming.org/>



Human Aspects



# Breaking the ice

- Difficult to start pairing
- The first sessions are awkward
- Pairs open themselves one to the other
- Afraid of criticism

# Behave Well

- Personal hygiene is important.
  - Take a shower (1<sup>st</sup> Rule of Pairing)
  - Use chewing gum
- Be polite!
- Have patience!
- Be honest!

“In pair programming  
wait 15 seconds before  
you point out mistakes.  
Give your pair a chance  
to find it first.”

(J. B. Rainsberger)

# Communicate Freely

- Focus on the code
- Ask questions
- Express things (even to a newbie)
- Talk about the code



# Anti Patterns

- Do not focus on typing/coding
- Do not hug the keyboard
- Do not dominate your pair
- Do not zone out (mobile phone etc.)



# Not for everyone

- Do I want to sit next to this person every day?
- In doubt raise your voice (don't risk the project)

# Getting Started



# Learning Pair Programming

- by doing
- first in safe place outside work
- Coding Dojo
- Code Retreat





# Practice Pair Programming



# Constraint: Ping-Pong

- A writes a test and sees that it fails.
- B writes the code needed to pass the test.
- A or B refactors the code as needed.
- B writes next test and sees that it fails.
- A writes the code needed to pass the test.
- And so on.

# Selling Pairing to Management

1. Pairing on or critical defects/problems (Problem Pairing)
2. Pairing on important business features
3. Pairing more often (Regular Pairing)
4. More often, etc.



A close-up photograph of a baby's face. The baby has light skin and blue eyes, which are wide open in a shocked or surprised expression. The baby's mouth is open, showing its tongue and teeth. The baby is wearing a light-colored, possibly pink, hooded sweatshirt. The background is dark and out of focus.

**MOAR PAIR  
PROGRAMMING!!1!**

**MOAR!!1!**

# When you work alone

- #PairWithMe, <http://www.pairprogramwith.me/>
- Remote Pairing, e.g.  
<https://www.softwerkskammer.org/groups/remotepairing>
- maybe substitutes
  - invite friend for review
  - use static code analysis

“Don’t be afraid of pair-programming - you’re not as good as you think, but you’re not as bad as you fear.”

(Ron Jeffries)



# Peter Kofler



# @codecopkofler

# [www.code-cop.org](http://www.code-cop.org)

# CC Images

- driver <http://www.flickr.com/photos/48694743@N00/517658329/>
- Hamster <http://www.flickr.com/photos/zebrapares/4529836138>
- 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/>
- rules [http://www.flickr.com/photos/mc\\_sensei/3935815039/](http://www.flickr.com/photos/mc_sensei/3935815039/)
- learn <http://www.flickr.com/photos/83633410@N07/7658268052/>
- sign <http://www.flickr.com/photos/hugosimmelin/2252095723/>
- ignition <https://www.flickr.com/photos/billselak/391759360/>
- dojo <http://www.flickr.com/photos/49715404@N00/3267627038/>
- Ping Pong <https://www.flickr.com/photos/jasperfields/2442146122>