

Pair programming for agile teams

Boyan Bonev - b.bonev@redbuffstudio.com



Topics

- What is it?
- Why try pair programming?
- An interlude with practical experience.
- Concerns & gotchas.
- Use as a management tool.
- Final thoughts.

What exactly is pair programming?

- Thesaurus to the rescue:

“*Pair programming consists of two programmers sharing a single workstation (one screen, keyboard and mouse among the pair).*

Agile Alliance

- Let's go informal:

It's actually about two developers **working** together.

The goals we strive for

...are still pretty much the same.

- Productivity.
- Quality.
- Knowledge sharing.
- Common ownership.
- Happy happy people.

GOOD CODERS...

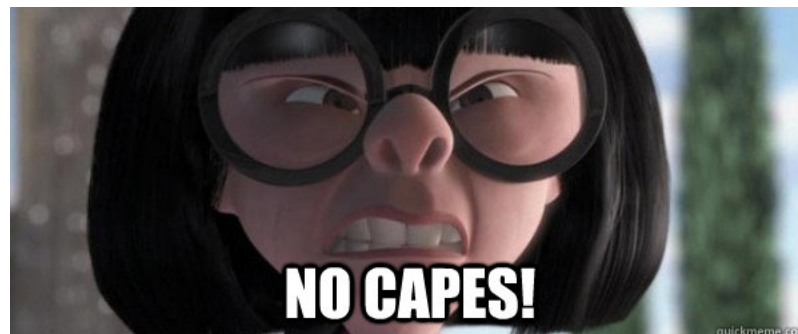


Once upon a time...

- About 4 years of pair programming so far.
- It's always harder when you start.
- Ticks all "Goals" on personal level.
- Have observed multiple "conversions"...
- ...but have also experienced the discomfort of pairing.
- Encouraged a culture of pairing in the current team.

So, you think you can ~~dance~~ pair?

- Good news, you are absolutely right, you **can** indeed!
- Suggest, don't mandate.
- Trial different setups.
- Listen to the feedback.
- Don't be the one creating the rules.



The Setup

Nice to have:

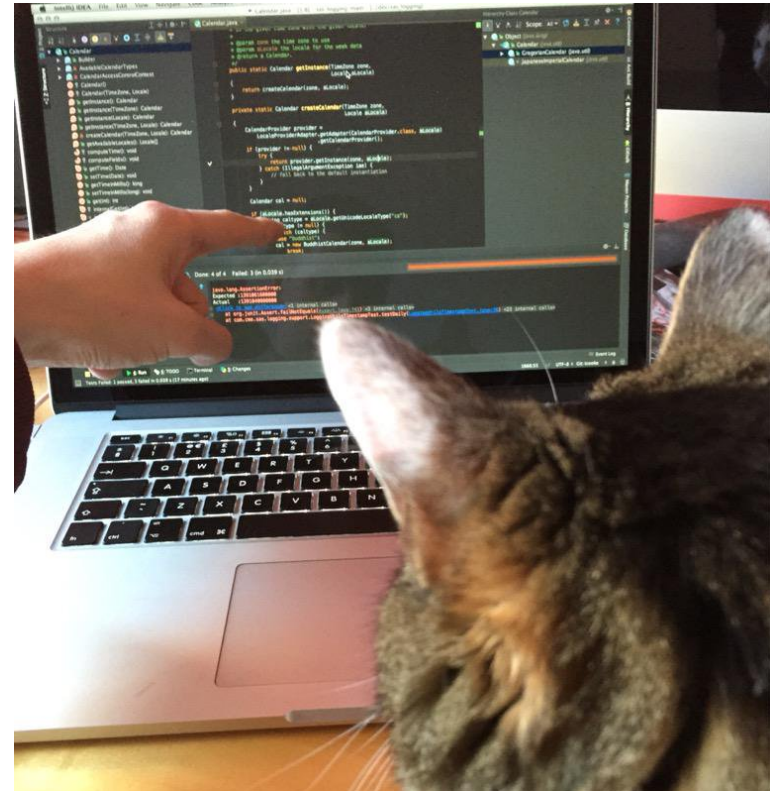
- Agile.
- Remote culture & tooling.
- Team culture of acceptance.

Worth trying:

- Different pairs.
- Different formats.
- Tool vs Practice.
- Use for timeboxing & experiments.

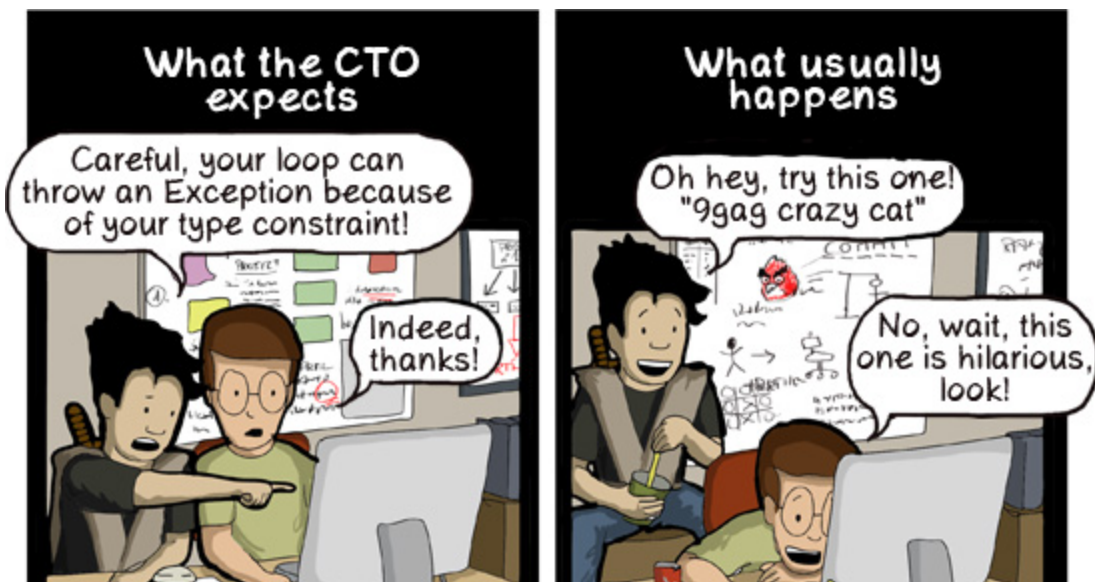
Pairing formats

- Constant pairs vs Story pairs.
- Peers vs Mentors.
- Compatibility vs Diversity.



Pairing process

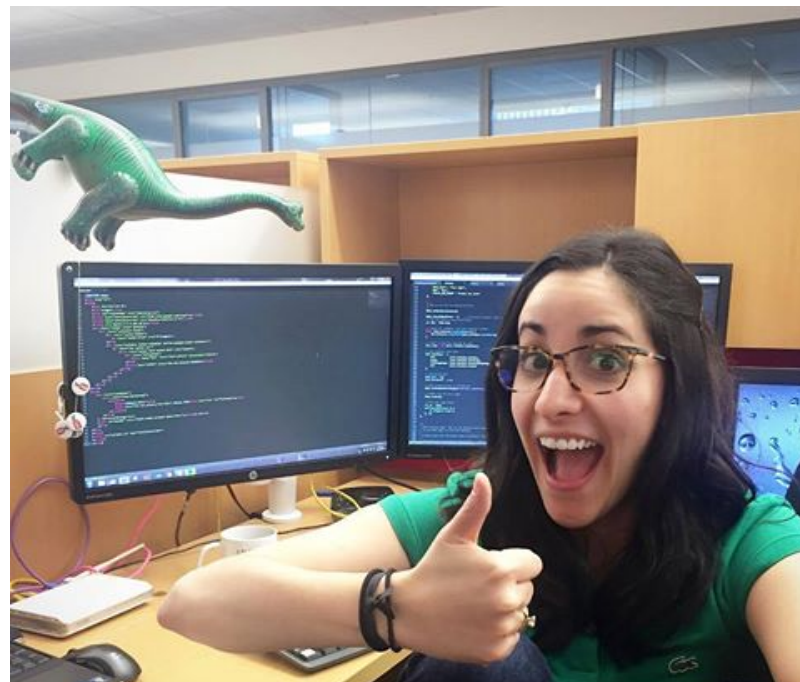
- Open space? Careful!
- Hotseat vs Remote.
- Lawful good vs Chaotic neutral.



100% guaranteed feedback

Great job, thumbs up, a++++,
will pair again...

...but why am I so tired?



Pair programming as a tool

- As part of the recruitment process.
- Extremely helpful for onboarding.
- Opens a possibility for conflict resolution.
- Great way to transfer and preserve domain knowledge.
- Stepping stone to building the company culture.

Why **shouldn't** we try pair programming?

- Negativity and scepticism.
- Slow, iterative environment.
- Red tape in the development process.
- Extreme communication difficulties.



The Verdict

- Try it, it's worth it.
- No need to stop there!
- Fun is good.

