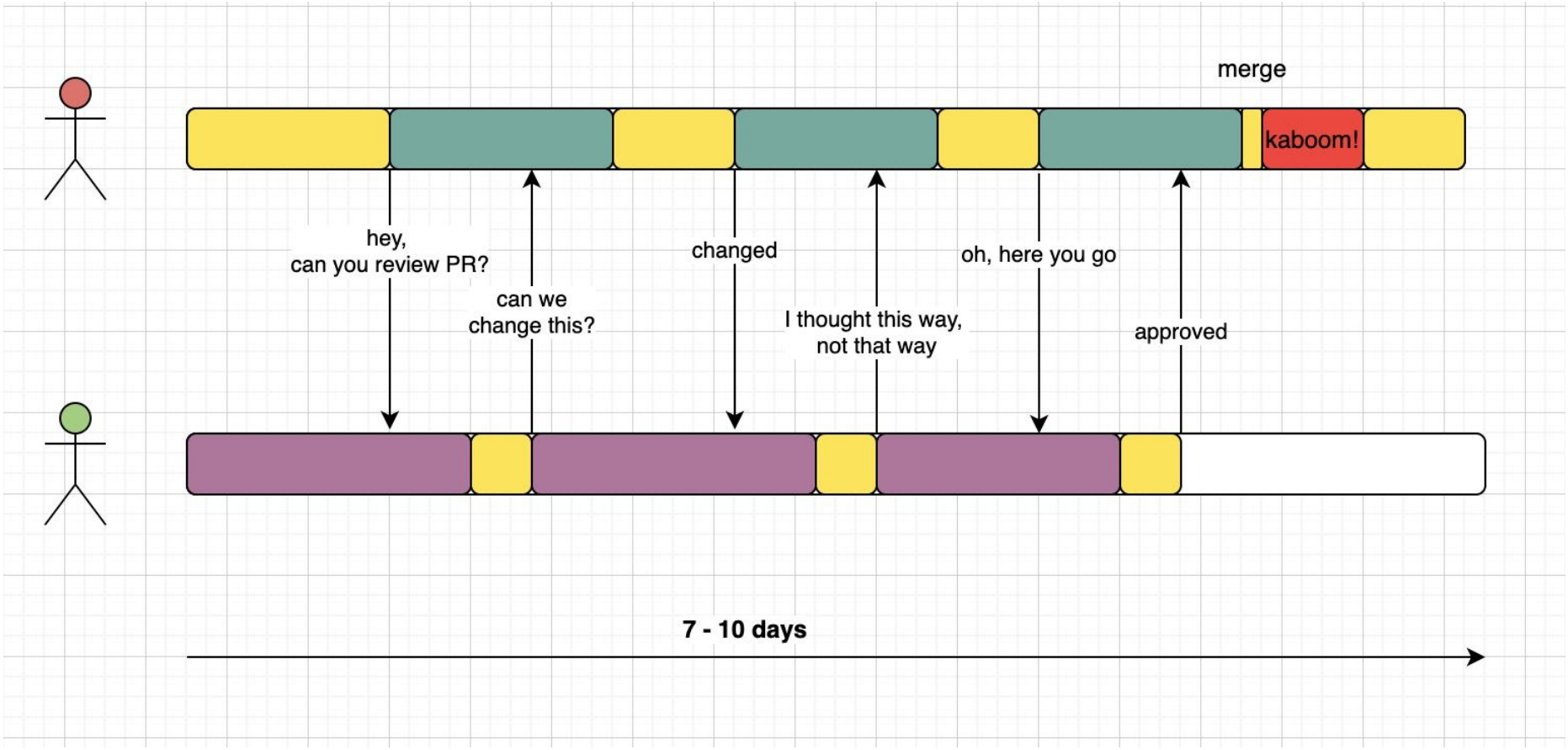
Pair Programming over Pull Requests

Dragan Stepanović
draganstepanovic.com
@d_stepanovic

How did we end up with code inspection at all?


merge
kaboom!
hey,
can you review PR?
can we
change this?
changed
I thought this way,
not that way
oh, here you go
approved
7 - 10 days

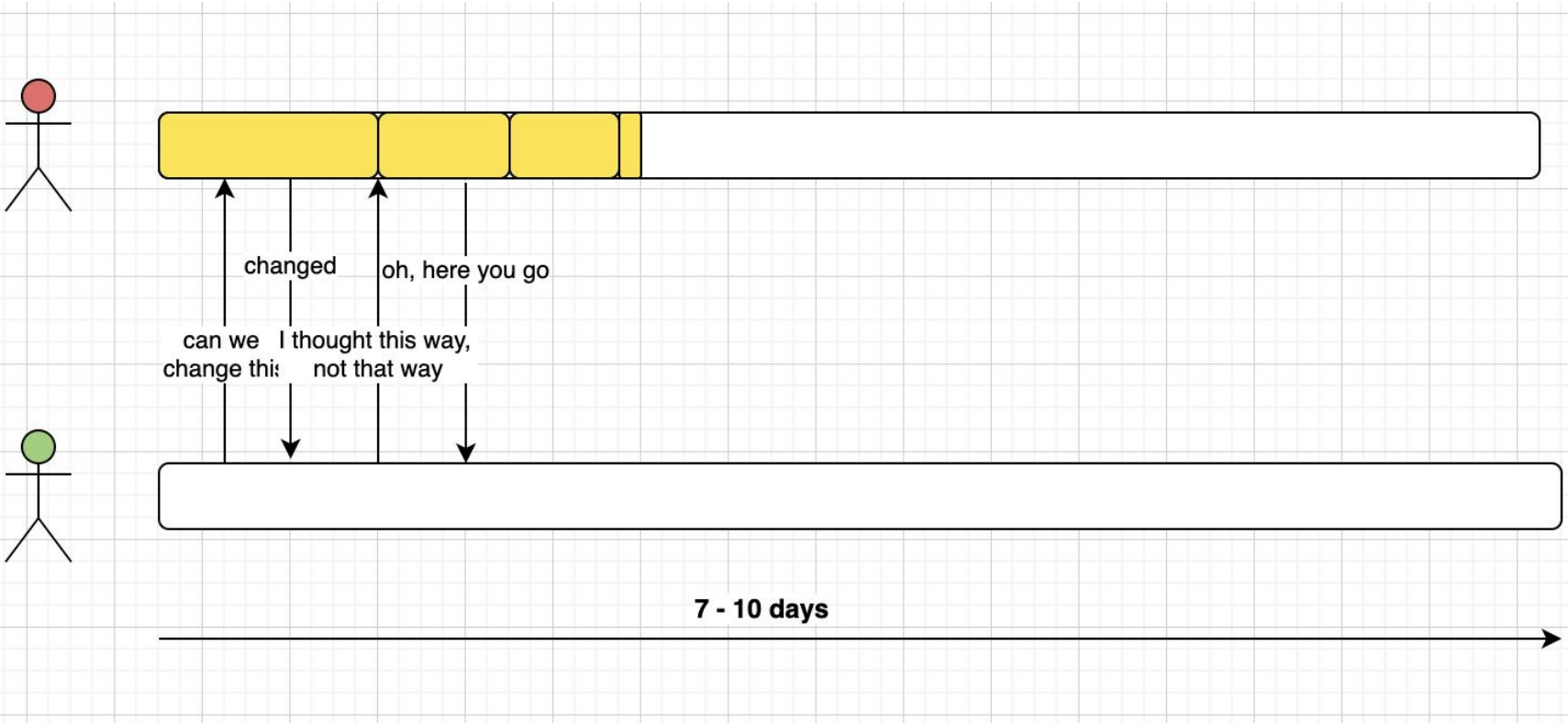
Lead time = processing time + queue (feedback) time

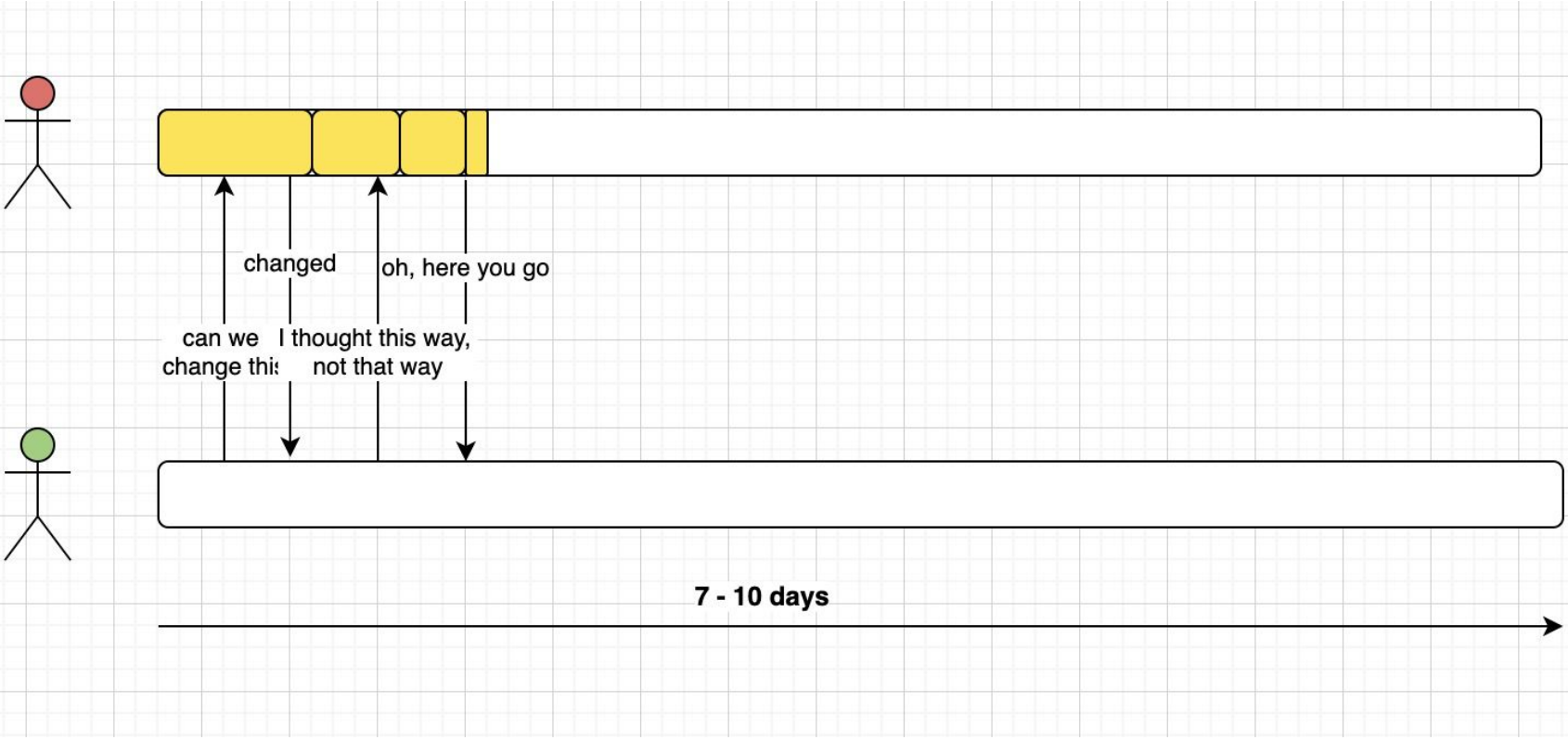
Shortening waiting time is way bigger leverage than shortening processing time

In order to achieve wealth optimize for minimizing losses over maximizing gains - Traders

Focus on the queue time!

Waiting time dominates processing time


changed
oh, here you go
can we change this
I thought this way, not that way
7 - 10 days


changed
oh, here you go
can we change this
I thought this way, not that way
7 - 10 days

Halve the PR size, then halve it again

What if we were to reduce every PR to a single line of code change?

Make your Pull Requests so small that you annoy reviewers so often that they just give up and join you in a pairing session instead

Get immediate feedback

Optimal PR size is one line code change that is reviewed immediately

---

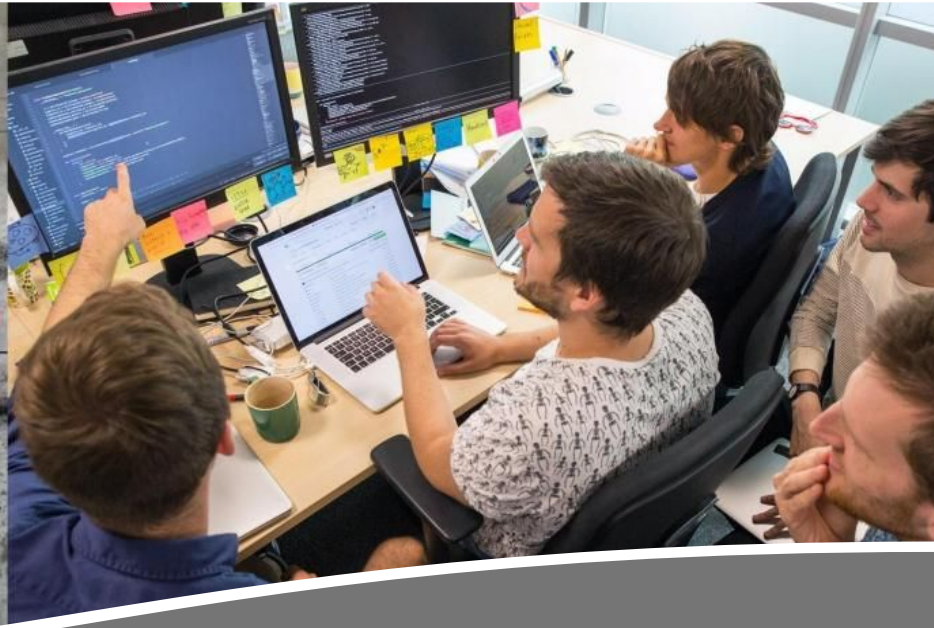
It's called Pair Programming

PRs were intended for open-source community

System drives behavior

“If I have to raise a PR for a simple method rename, I’d rather not do it”

Pairing drastically reduces cost of review



Move the work to the people

- **Developers start to protest the code review process.** If a reviewer only responds every few days, but requests major changes to the CL each time, that can be frustrating and difficult for developers. Often, this is expressed as complaints about how “strict” the reviewer is being. If the reviewer requests the *same* substantial changes (changes which really do improve code health) but responds *quickly* every time the developer makes an update, the complaints tend to disappear. **Most complaints about the code review process are actually resolved by making the process faster.**

https://google.github.io/eng-practices/review/reviewer/speed.html

Reviewing changes
10 lines - WTF?!
500 - LGTM

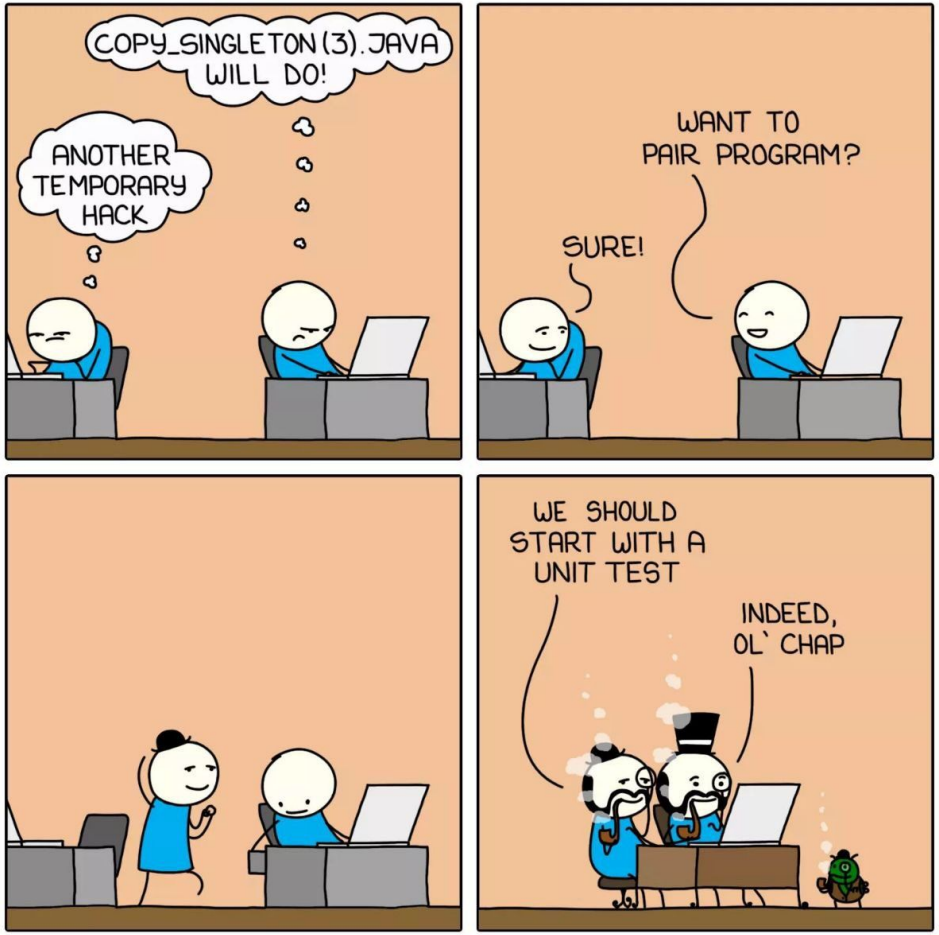
Quality theater

System inhibits building quality in

PR metrics

PAIR PROGRAMMING

MONKEYUSER.COM



Way more engagement of team members

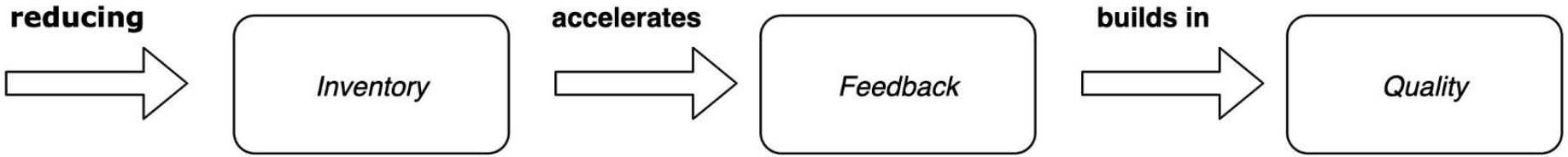
Code Review → critique
Pairing → collaborative

https://www.infoq.com/articles/adopting-pair-programming/

System incentivizes increase in batch size

Pairing guarantees small batches


LEFT
EXIT 12
BEST SOLUTION
MOST EMOTIONALLY INVESTED SOLUTION
LONG FEEDBACK LOOP
imgflip.com



https://dragan-stepanovic.github.io/2019/12/13/reducing-inventory-builds-quality-in.html

Different (time) contexts makes it harder to understand each other and have shared (valid) assumptions

Written word is way harder to not be interpreted as not intended

Text as a medium is very expensive compared to spoken word

Spoken word is rich, and it elevates understanding

Blame the system, not the people.

#NoEvents

PR cons

- every new review of the same PR takes context switch and longer time to understand the context
- "you solved the wrong problem" or "you solved it in a wrong way"
- the bigger the PR, the longer review and rework time
- delayed feedback causes stronger oscillations and longer lead times
- too many approvals without comments/changes

PR cons

- Incurs technical debt
- more merge conflicts because of high WIP
- not understanding the context and having to wait for answers
- sometimes people avoid commenting back

Breaking the knowledge silos

Like a never-ending knowledge sharing session

Seniors never get bored. Why? Because they are mentored all the time.

Dragan Stepanović
@d_stepanovic

What if senior developers were not allowed to write a single line of code alone? Everything has to go through the hands of less experienced developers.

How would this enabling constraint change the world and dynamics in your team(s)?

10:56 PM · Jun 25, 2020 · Twitter Web App

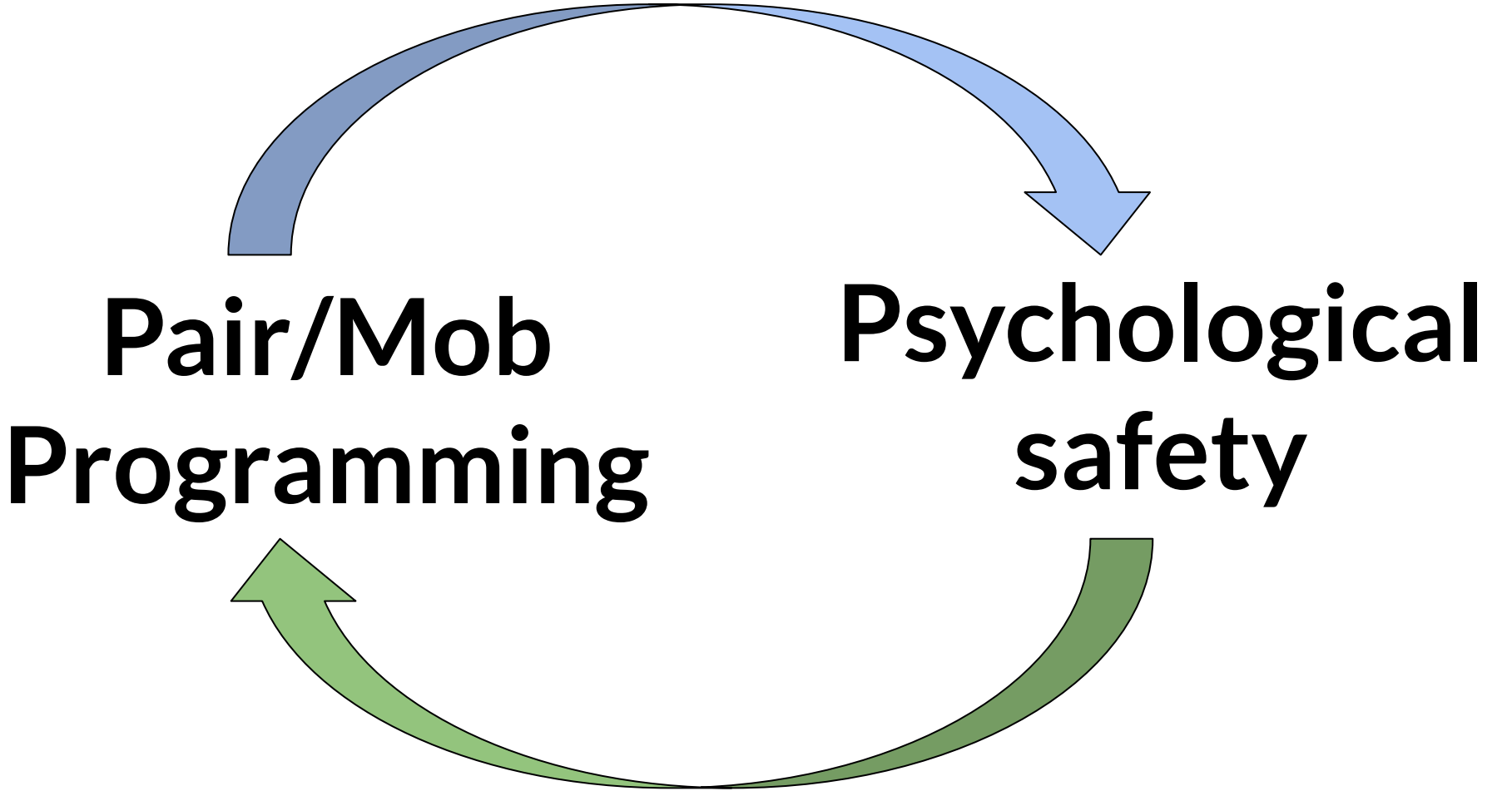
View Tweet activity

48 Retweets and comments **217** Likes

Whole is greater than the sum of its parts

PP helps build the trust

More focus
Less interruptions


Pair/Mob Programming
Psychological safety

Limiting WIP out of the box → Higher throughput

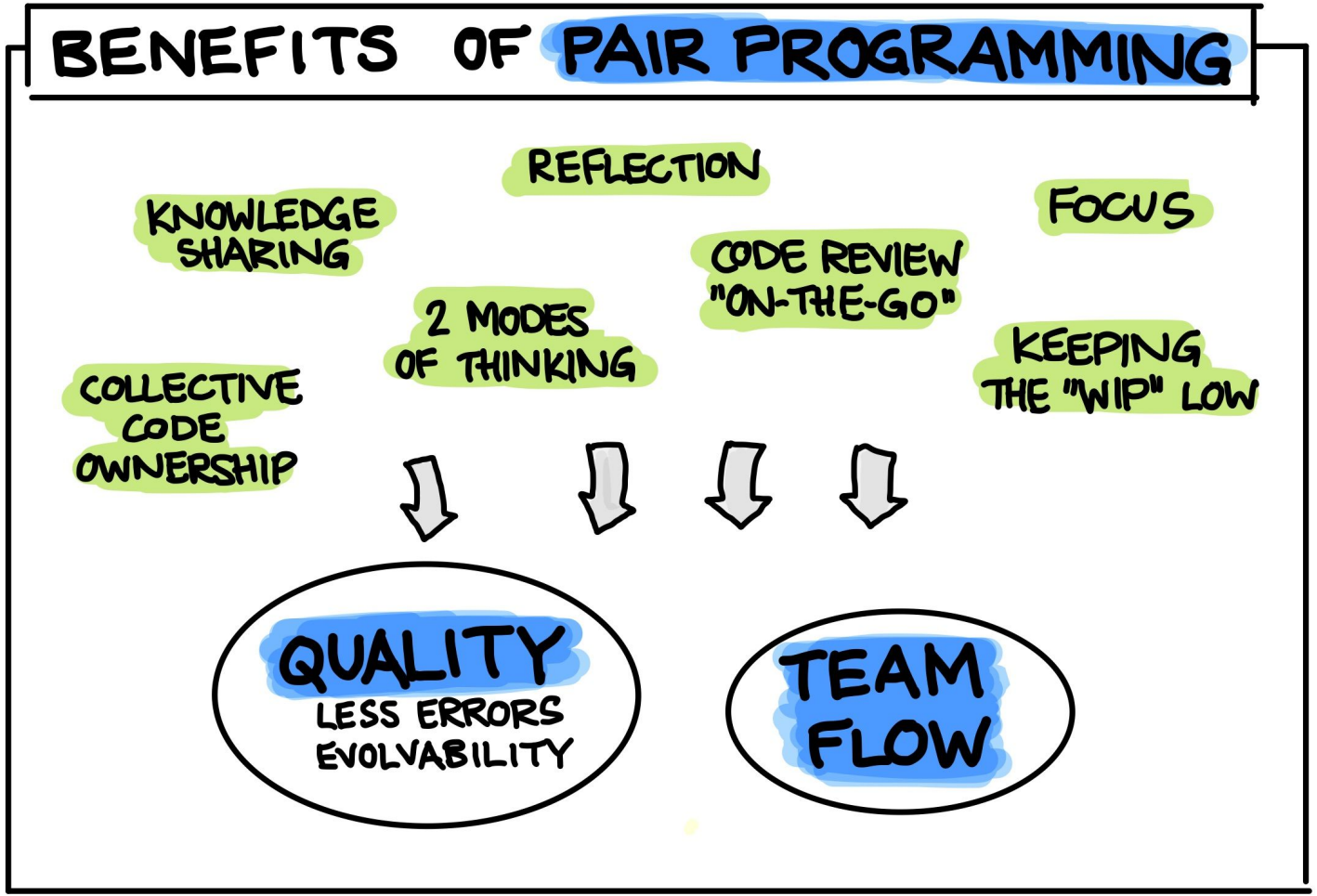
Pairing exposes team dynamics which can be a candidate for improvement

Collaboration is a skill that takes time to learn

Pair programming

"The adjustment period from solo programming to collaborative programming was like eating a hot pepper. The first time you try it, you may not like it because you are not used to it. However the more you eat it, the more you like it."

Anonymous XP practitioner



https://martinfowler.com/articles/on-pair-programming.html

Modes of PP

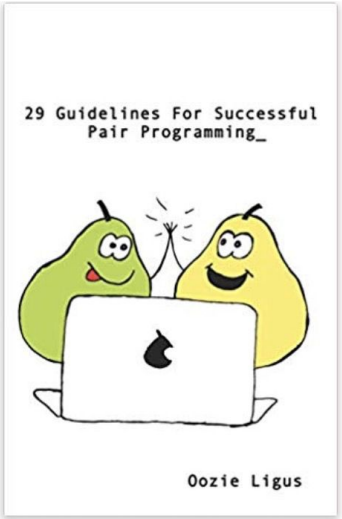
- Driver - navigator
- Silent
- Promiscuous
- What if seniors in the team were not allowed to code on their own, but instead every idea was to go through junior’s hands?

How to try it out?

- Incrementally
  - An hour per day
  - Phrase it as an experiment and reflect after one or two weeks
  - What did you like, did not like, learnings, surprises?
- Try it all the time for a short period of time
  - One week
  - In order to change mindset, we cannot **think out** of the current mindset, because if we could we would already be there. But we need to **act away** of it. Sometimes it's not feasible to incrementally change the mindset (move to the other end of the spectrum).
    - REBT
    - Changing the culture through change of behavior

When not to PP?

How to start pairing?



- How to start Pair Programming - Amitai Schleier
- What is Pair Programming - Sarah Mei
- Experience from about 8 years of pair programming
- Pair Programming Antipatterns
- and some more
- On Pair Programming and a video talk
- How to Pair Program