



Remote Mob Programming

Für Team-Zusammenhalt in Homeoffice-Zeiten



Thomas Much

 @thmuch

04.05.2021

Softwareentwickler

Agile Developer Coach



Mob Programming
seit 2016

Remote Mob Programming
seit 2020

... auf den Schultern von Riesen ...

 @thmuch



Mob Programming

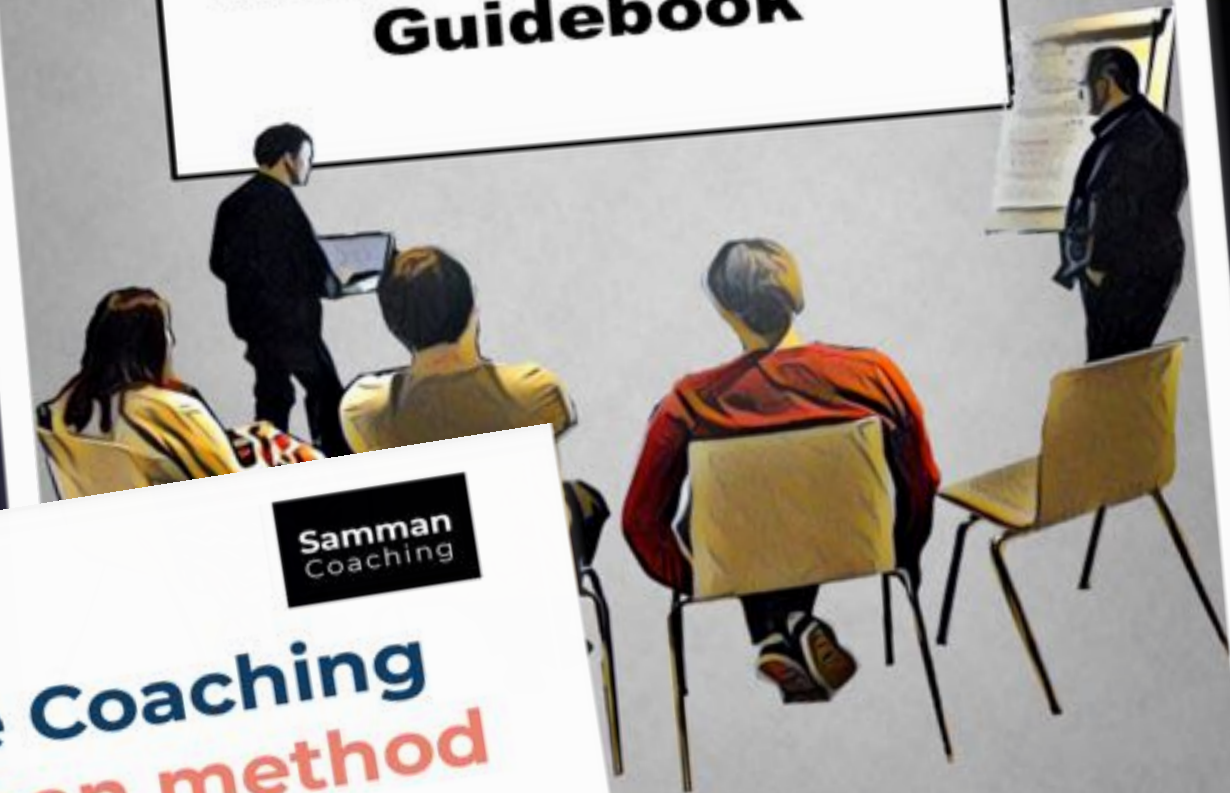
A Whole Team Approach



By Woody Zuill and Kevin Meadows



The Mob Programming Guidebook



MOB PROGRAMMING GUIDEBOOK

MAARET PYHÄJÄRVI

Technical Agile Coaching with the Samman method

By Emily Bache
Foreword by Kent Beck



Remote Mob Programming

At home, but not alone

INNOQ

Foreword by Mark Pearl

The Pragmatic Programmers

Pragmatic express

Code with the Wisdom of the Crowd

Get Better Together with Mob Programming

Mark Pearl
edited by
Tammy Coron





Marcus Hammarberg
@marcusoftnet

#MobProgramming gets another well deserved shoutout and excellent quick intro by @lisihocke #agrs2018

This is the most important improvement I've seen the last couple of years.

Please get hold of me if you want to
Especially if you think it's ineffective

[Tweet übersetzen](#)

11:33 vorm. · 20. Sep. 2018 · [Twitter for iPhone](#)



Allen Holub
@allenholub

4/6 I've had a lot of experience with mob programming. I've introduced it to many teams, run workshops, seen its impact. It's one of the most significant improvements that a team can adopt. Most of the teams I've worked with love it, though not all.

[Tweet übersetzen](#)

12:24 vorm. · 23. Aug. 2019 · [Little Pork Chop](#)

XP PragProg

Agile Scrum

Software Craft

IT-Kanban

~ 20 Jahre alt

User Stories

Pair Programming

Refactoring TDD Iterations

Continuous Integration

Mob Programming

~2011 eher nebenbei entstanden

Seit 2014 weltweit diskutiert & eingesetzt

(ältere Ideen haben sich nicht durchgesetzt)



Woody Zuill

<https://woodyzuill.com/>

<https://twitter.com/woodyzuill>

Mob Programming

„Mobbing“ 🤔🤨

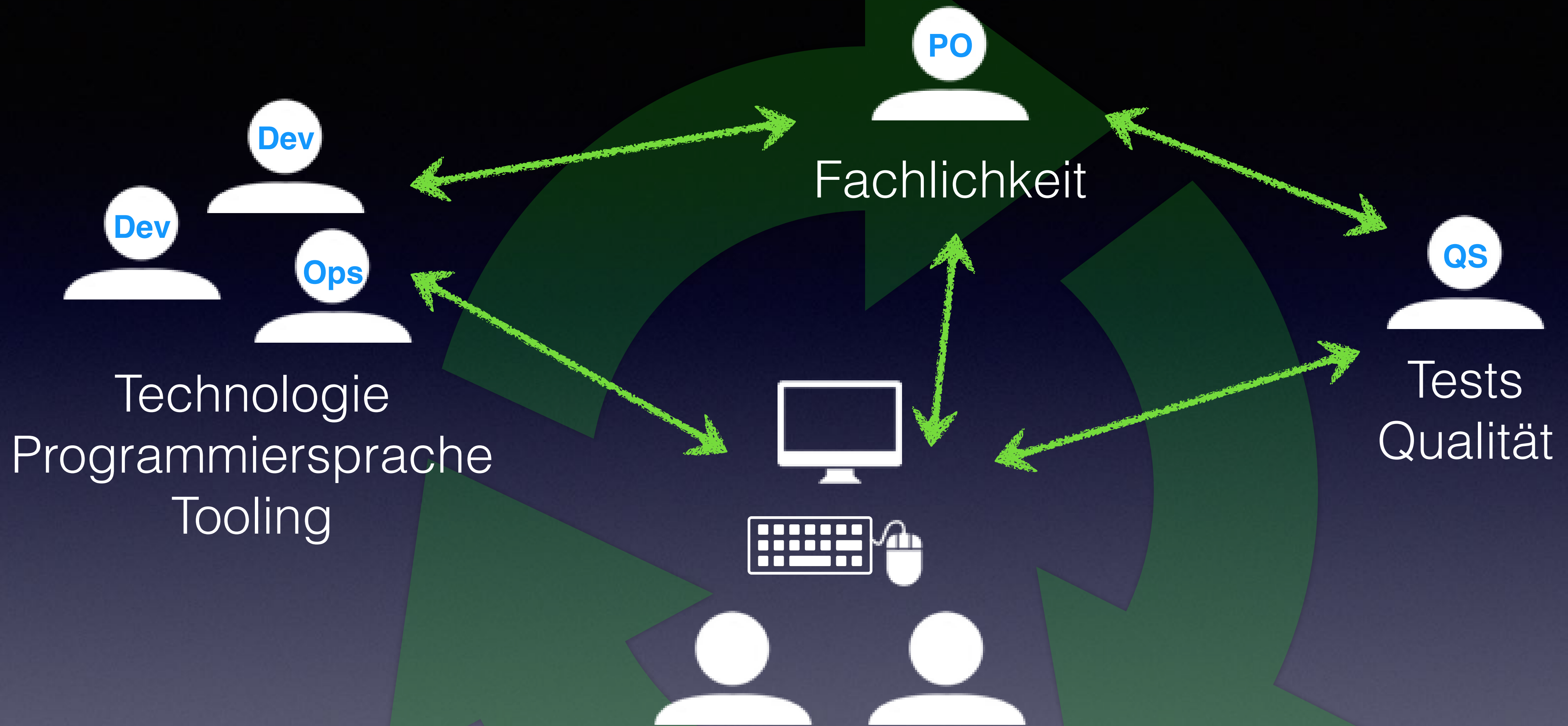
Group Programming
(Whole) Team Programming

Ensemble Programming*

„Mobsen“** 😊

*) <https://visible-quality.blogspot.com/2020/05/five-years-of-mob-testing-hello-to.html>

***) Credit: Team „Transformers“ @ OTTO



"**All the brilliant minds** working together on the same thing, at the same time, in the same space, ~~and at the same computer.~~"

– Woody Zuill

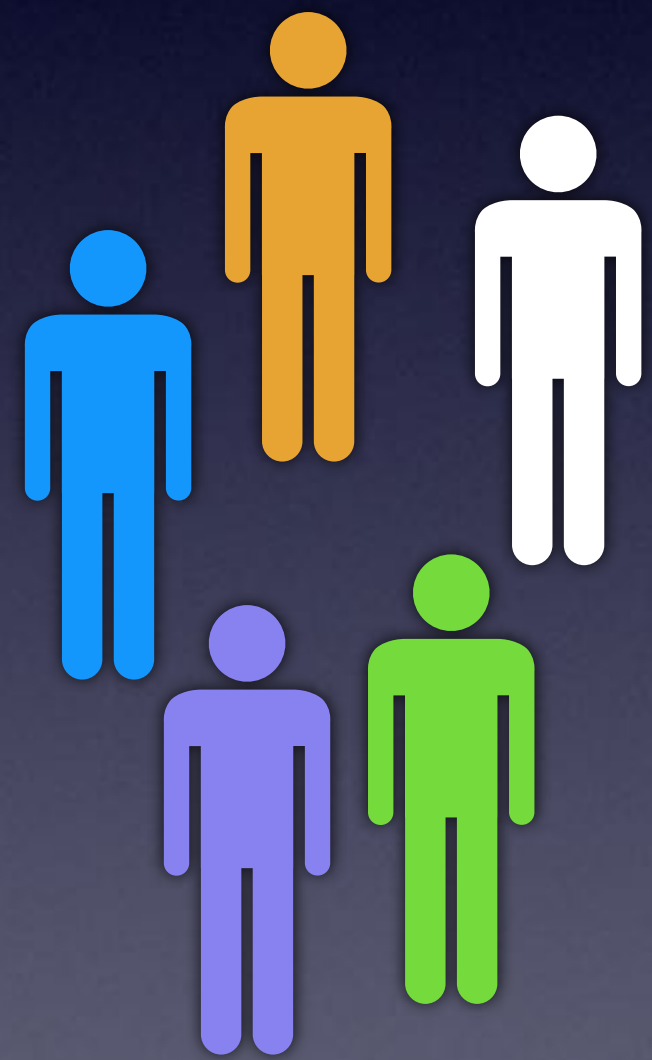
Wie arbeitet Dein Team? (Ist das ein Problem?)

Wie funktioniert (Remote) Mob Programming?

Warum funktioniert das (& löst diverse Probleme)?

Wie arbeitet Dein Team?

Sind wir ein Team?



Ein 5er-Team

Oder eher mehrere „Teams“?



Fünf Teams von Einzelnen

Oder eher mehrere „Teams“?



Fünf Teams von Einzelnen

5 Devs für 1 (EINE) Aufgabe???

Wofür optimiert das Management?

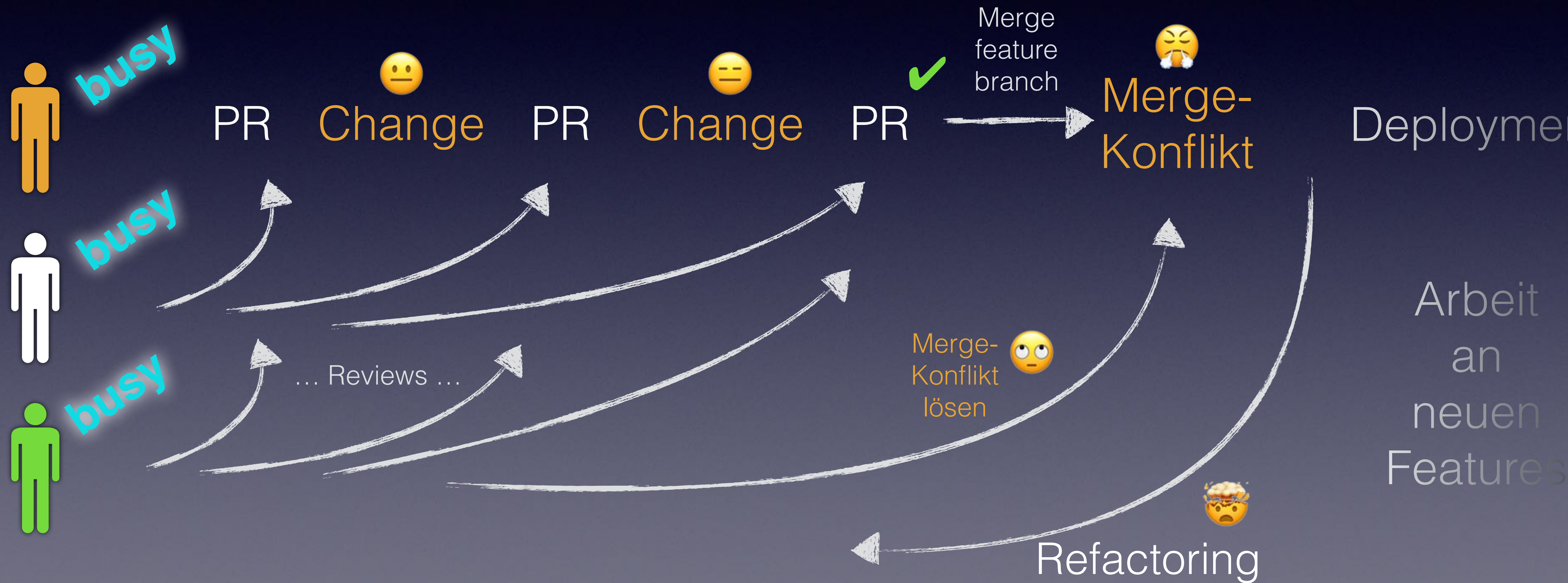
Wichtigstes Feature benutzbar zum Kunden ausliefern?

oder

Alle Entwickler möglichst gut auslasten / beschäftigen?

(Kleiner Hinweis: Genau das bekommt Ihr dann auch.)

Warten, warten, warten ...

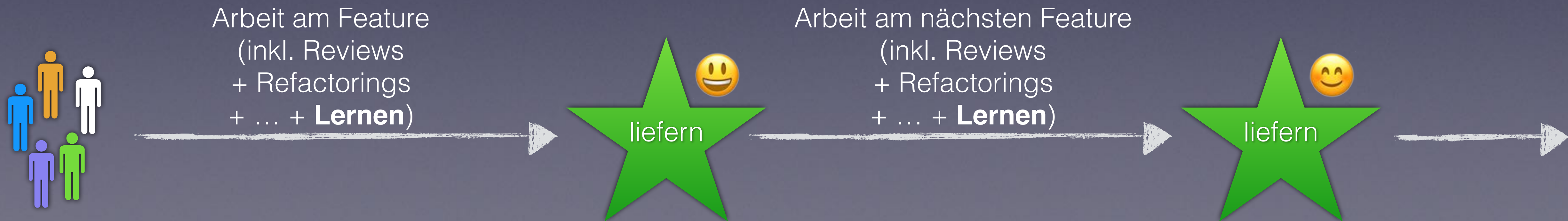


Flow

Kontinuierlich Wertvolles an unsere Kunden ausliefern

Anzahl der erledigten Dinge maximieren

Verschwendung minimieren



Aber die Geschwindigkeit!

„~~busy~~“ sein → Zeug ~~raushauen~~

„If you need to be fast, **slow down**“

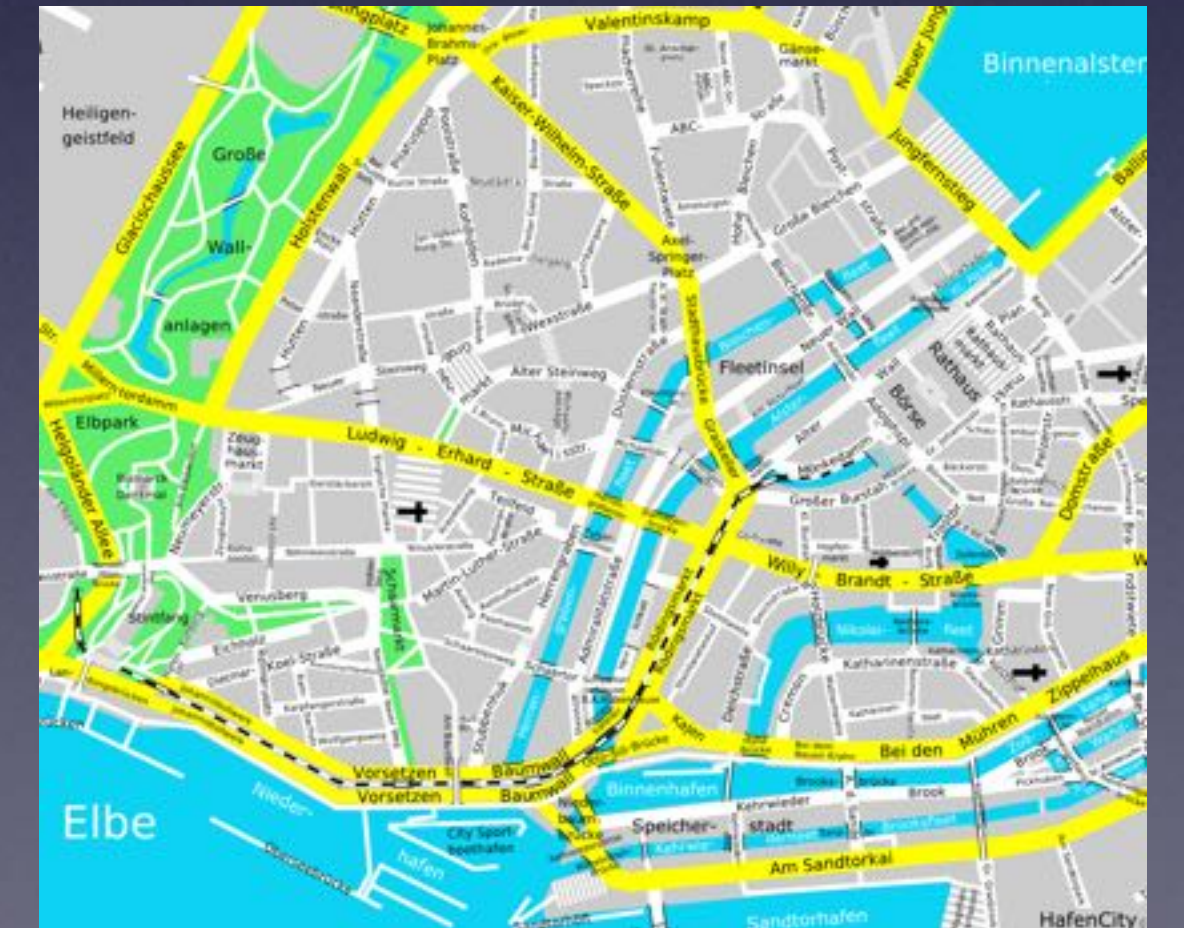


Seid nicht langsam.
Aber vermeidet Hektik.

Wie funktioniert (Remote) Mob Programming?

Wichtig:
Es gibt kein „Richtig“
oder „Falsch“!

Driver & Navigator



Driver-Observer Anti-Pattern



Worker

Watcher

Aufmerksamkeit & Kreativität

Rollenwechsel!

**Tastaturwechsel nach fester Timebox
(alle 5-10 Min. / remote 10-15)**

Timer App, z.B.

* <http://mobster.cc/>

* <https://cuckoo.team/>

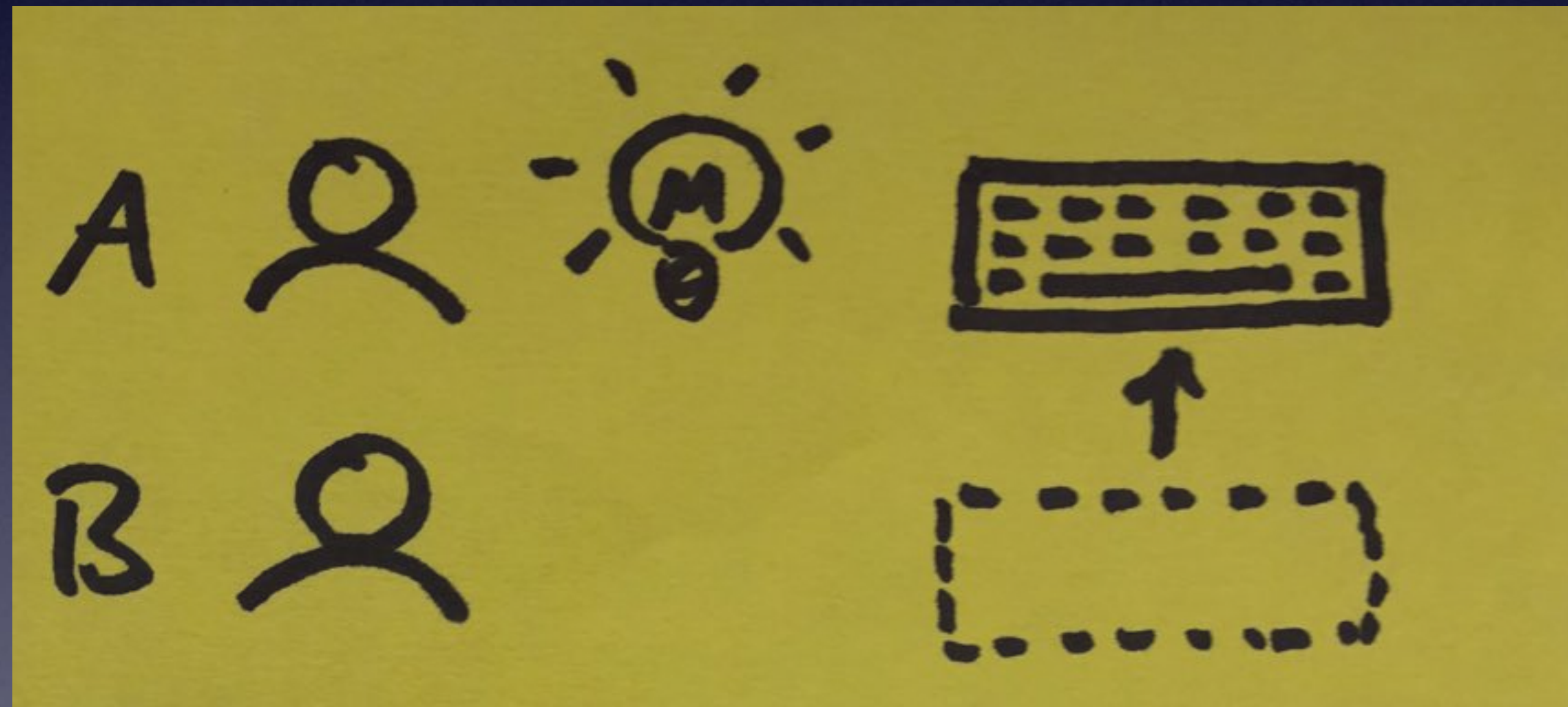
* <https://mobti.me/>

oder *Smartphone-Timer*

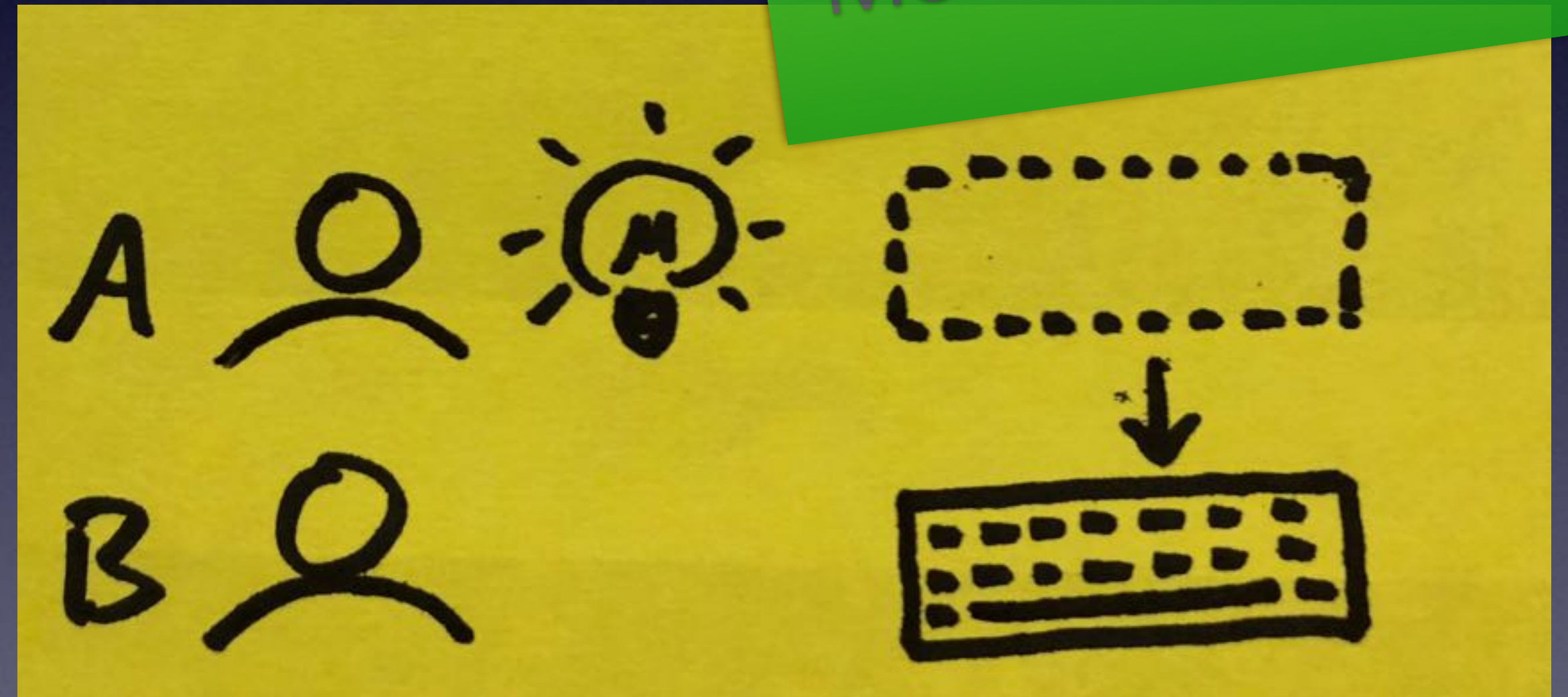


Strong Style Navigation!

„klassisch“



„Strong Style“ beim
Mob Programming



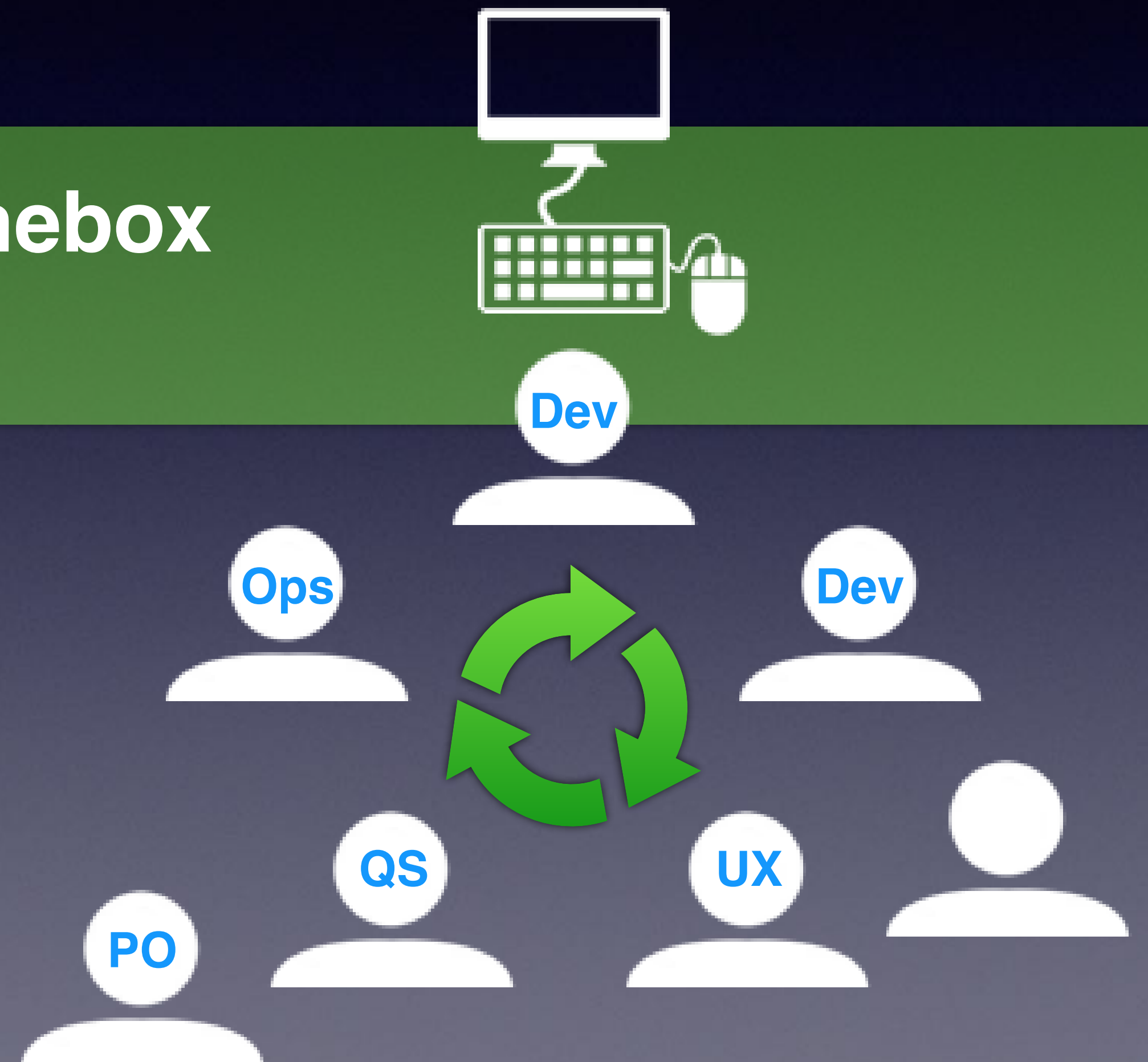
@LlewellynFalco

„All the brilliant minds“


Rollenübergreifend!

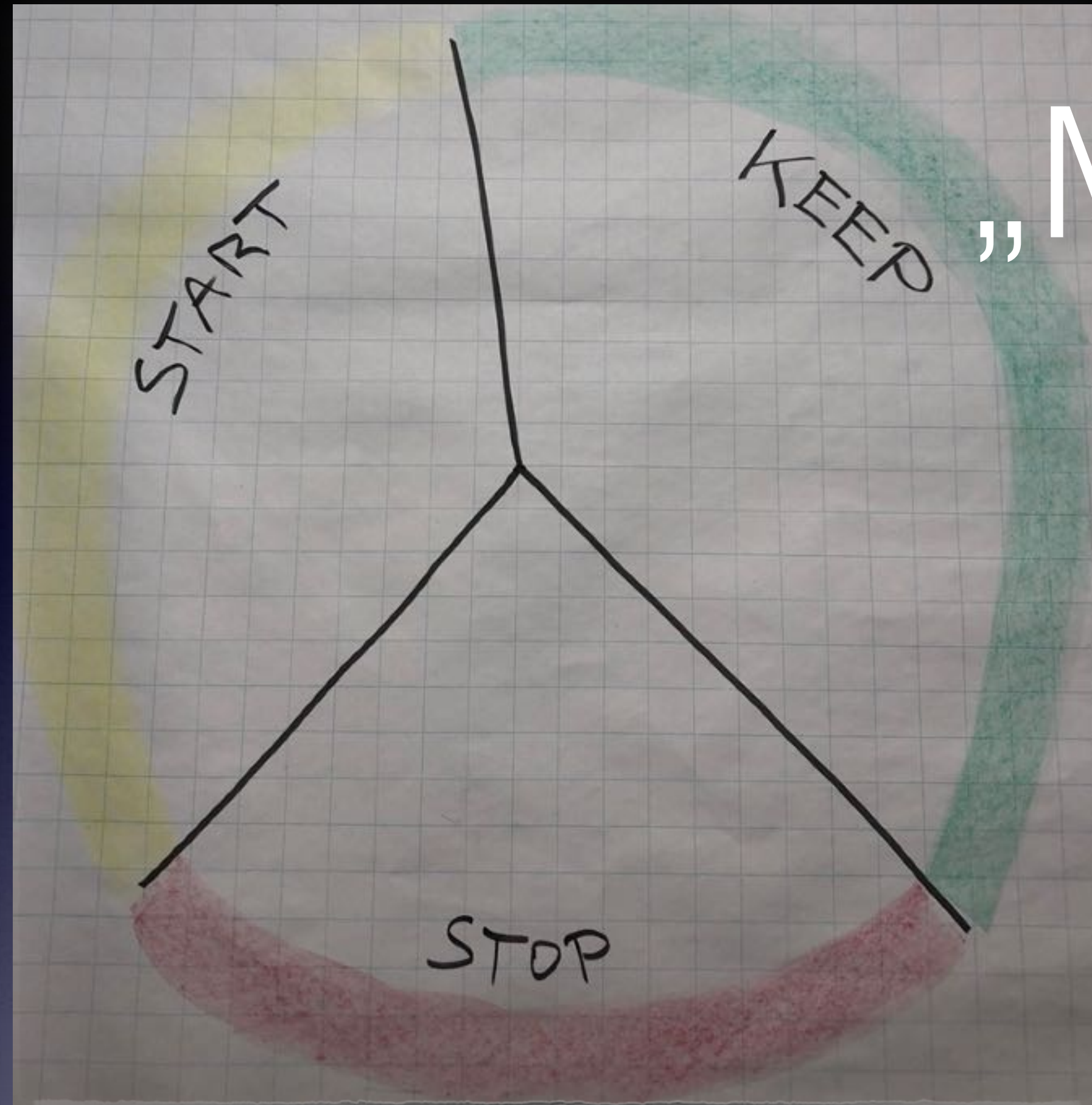
**Tastaturwechsel nach fester Timebox
(alle 5-10 Min. / remote 10-15)**

Die **wichtigste** Aufgabe
zügig und gut fertig bekommen.



Ablauf

- Regeln für Mob-Session festlegen (Timebox, Navigator? etc.)
- Kurze Intro zur Problemstellung
- Discovery/~~Estimation~~/Planning/Implementation 
- Pause ca. 1x pro Stunde
- Dauer je Mob-Session? 2h, ..., 1/2 Tag, 1 Tag
- Kurze Retro



„Mini-Retros“

Title, Topic, Subject (Participants?) etc.

KEEP

START

STOP

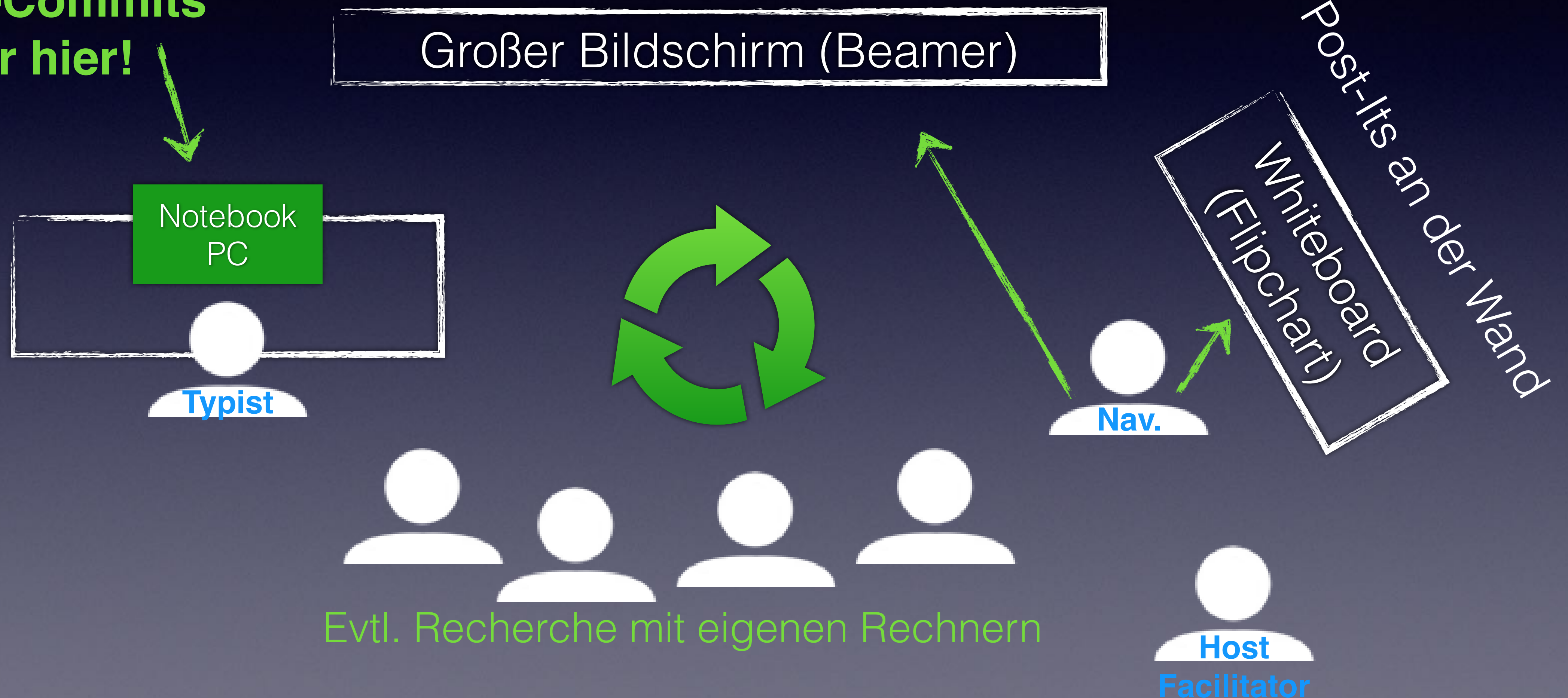
Next steps for tomorrow / for next session:

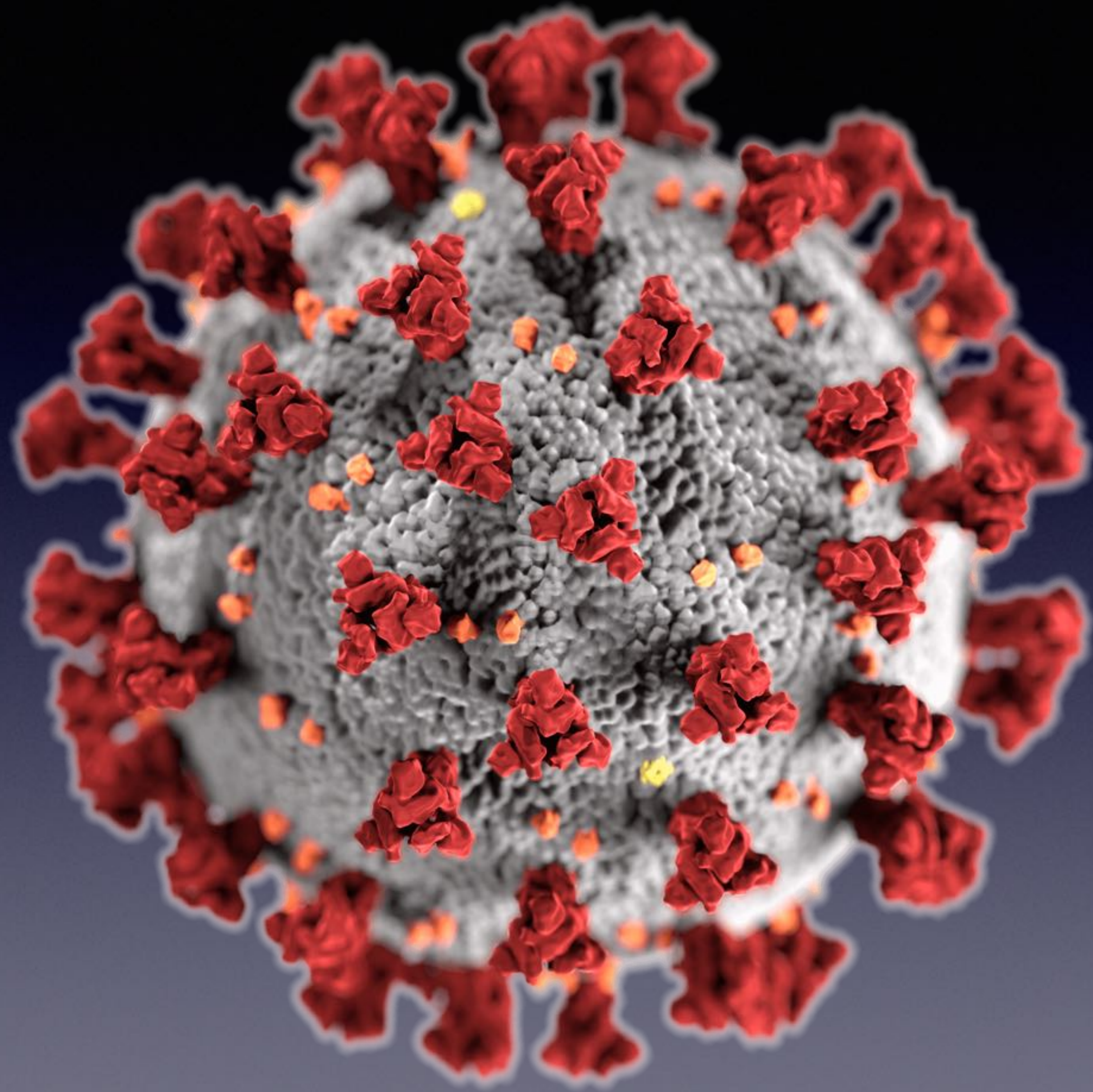
Fokus auf "Keep" & "Start" ⇒ **"Turn up the good"**

Kurzen, knappen "Check-In" für nächste Session schreiben

Typist, Navigator & Rest vom Mob

Code-Commits
nur hier!





<https://phil.cdc.gov/Details.aspx?pid=23312>



Remote Mob Programming

Remote Everybody
Camera Always On
Regular On-Site Meetings
Small Team
Same Time
Typist and the Rest of the Mob
Screen Sharing
10 Minute Intervals
Git Handover
Group Decisions
Constant Momentum
Learn from the Team
Trust
Save the Planet
Dine with your Family



<https://www.remotemobprogramming.org>

Remote Mob Programming

Anders als „normale“ verteilte Teams!

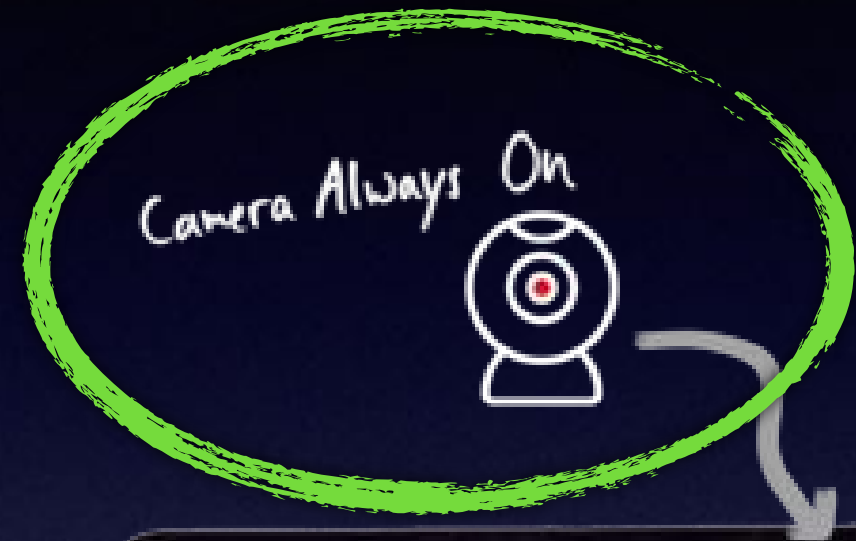
(GitLab) Remote Manifesto ↔ Remote Mob Programming

„Asynchrone Kommunikation
mehr als
synchrone Kommunikation.“

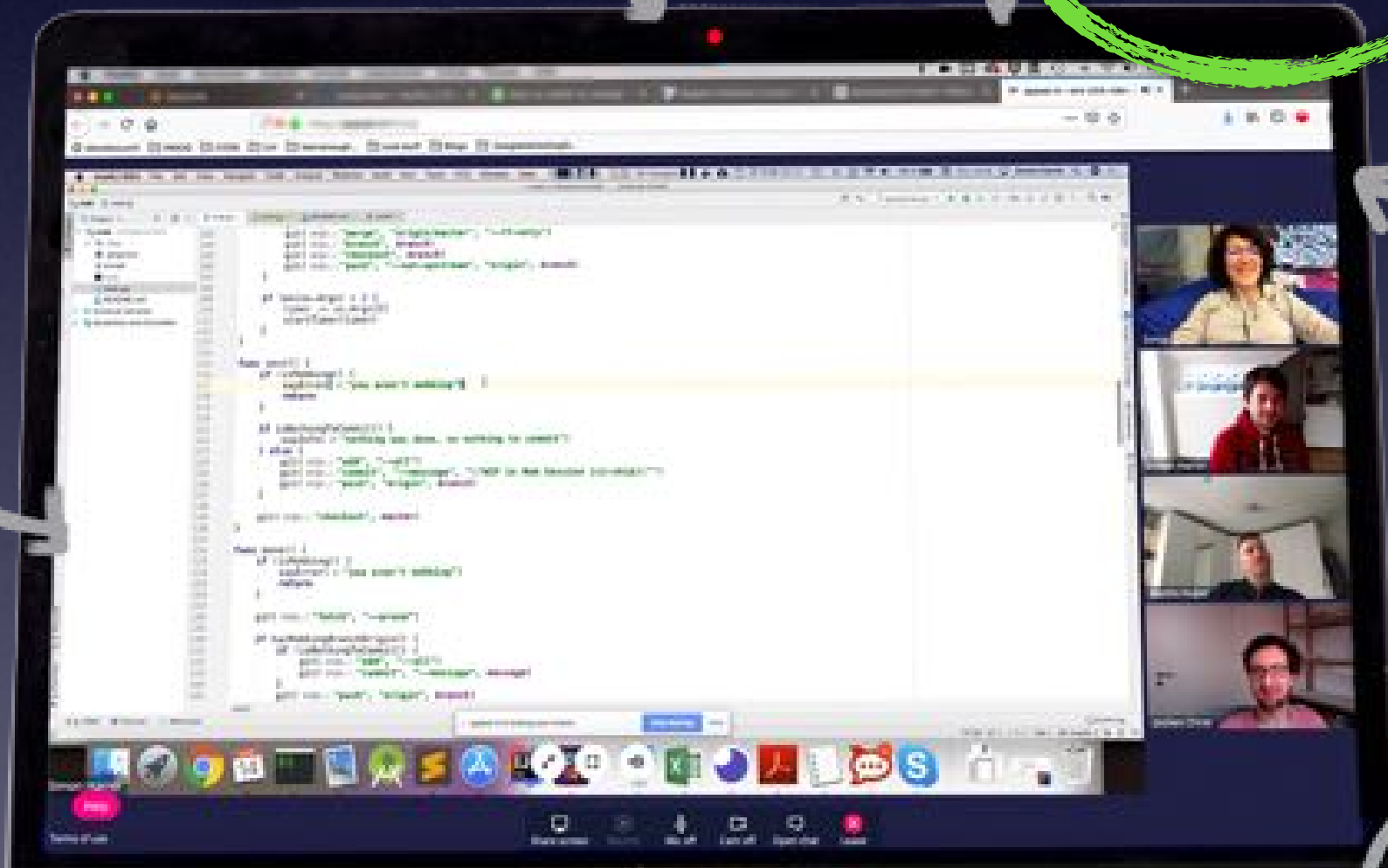


„zeitgleich“

Wir sehen uns!



10 Minutes Intervals



Typist and the Rest of the Mob



Dine with your Family



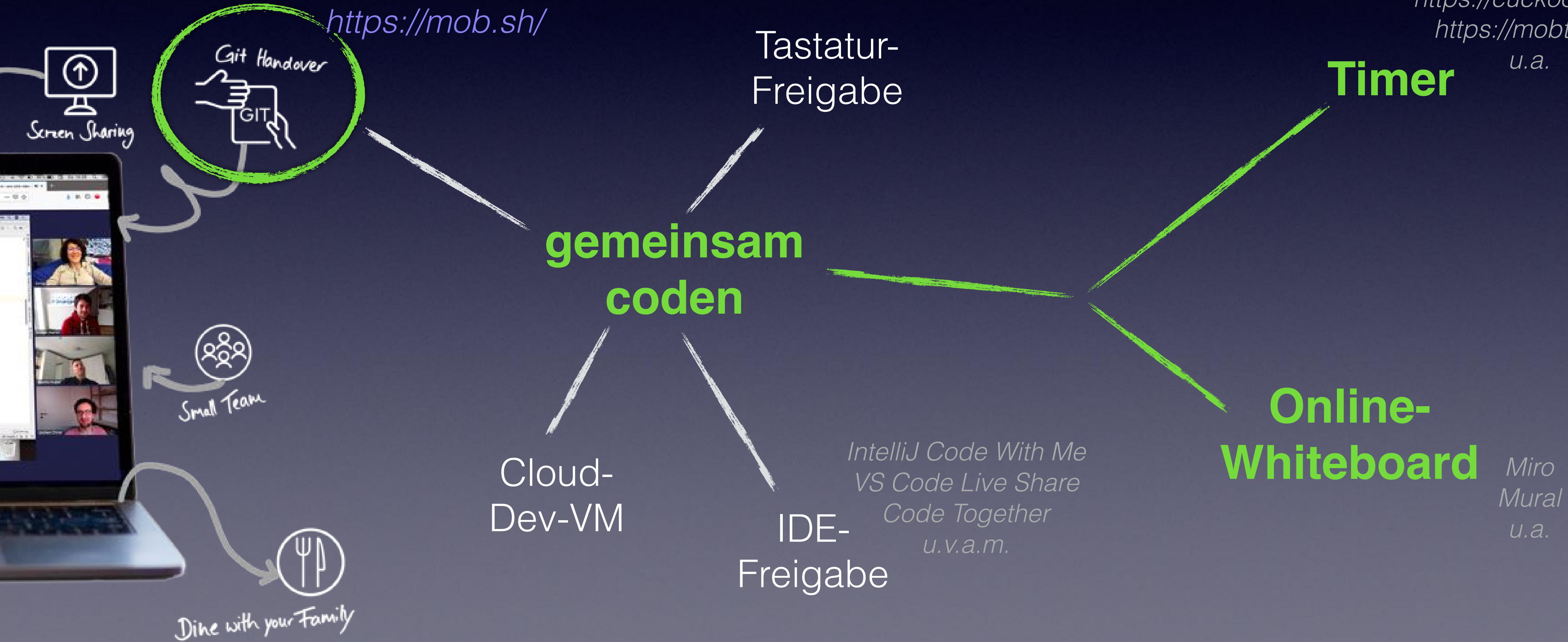
„alle remote“

Wir entwickeln zusammen!



<https://mob.sh/>

<https://cuckoo.team/>
<https://mobti.me/>
u.a.



Tastatur-Freigabe

Timer

gemeinsam coden

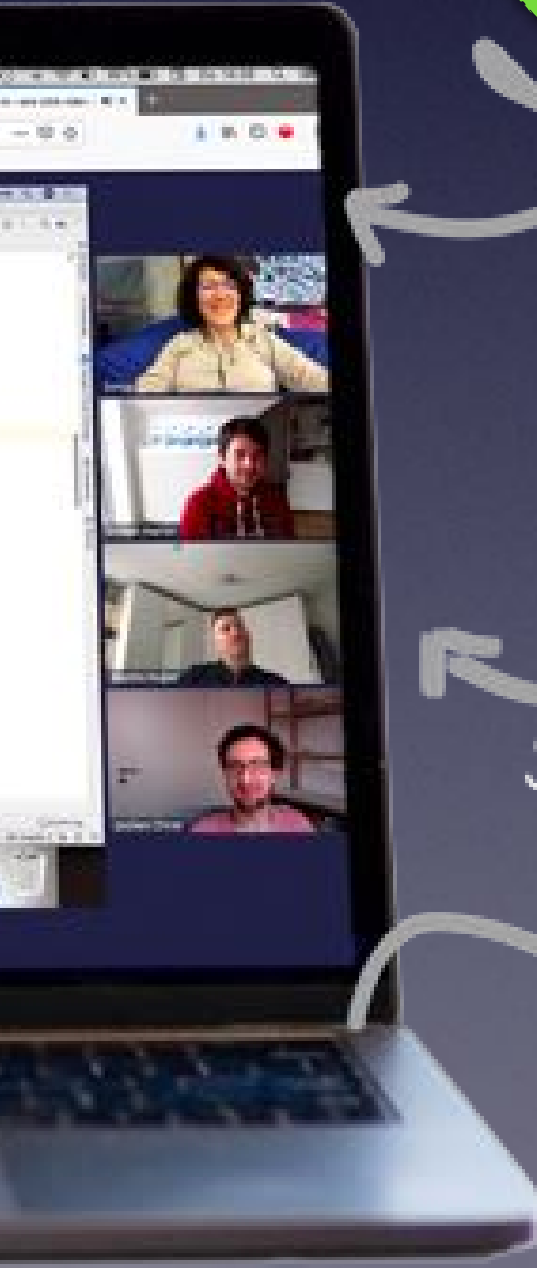
Online-Whiteboard

Cloud-Dev-VM

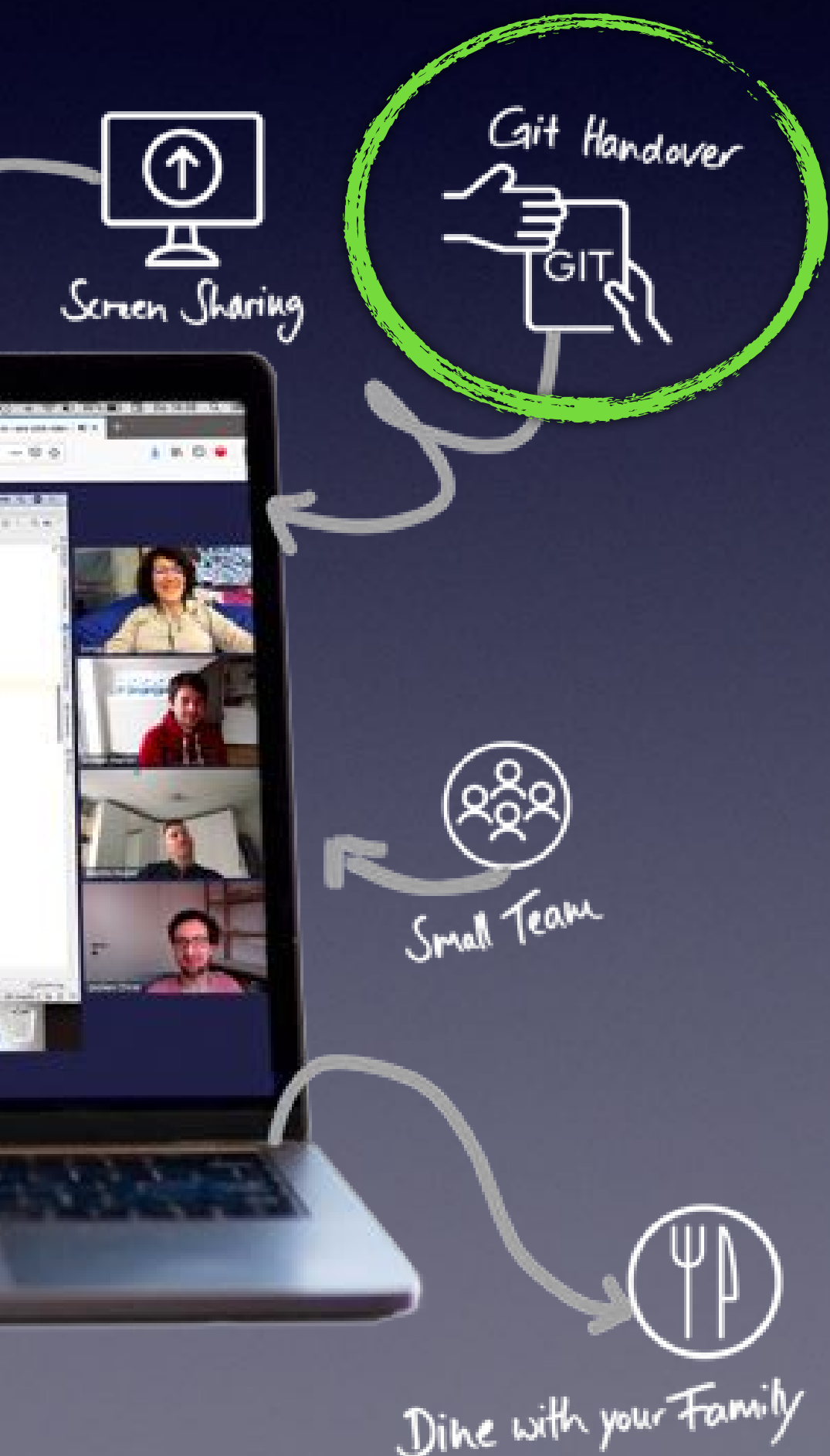
IDE-Freigabe

*IntelliJ Code With Me
VS Code Live Share
Code Together
u.v.a.m.*

*Miro
Mural
u.a.*



Gemeinsam coden mit Git-Handover



```
# simon begins the mob session as typist
simon$ mob start 10
# WORK
# after 10 minutes...
simon$ mob next
# carola takes over as the second typist
carola$ mob start 10
# WORK
# after 10 minutes...
carola$ mob next
simon$ mob start 10
# WORK
# After 6 minutes the work is done.
simon$ mob done
simon$ git commit --message "describe what the mob session was all about"
```



<https://mob.sh/>

IntelliJ plug-in: 
<https://plugins.jetbrains.com/plugin/14266-mob>

Wo kann ich mithelfen (oder lernen)?

The screenshot shows a Microsoft Teams chat window for a team named "T - Mob Sessions". The chat history includes several messages:

- A message from "PD" stating "PDN-1100 Initiales Deployment gestartet" (PDN-1100 Initial deployment started), with 2 replies and a meeting duration of 37 minutes and 59 seconds.
- A message from "PD" stating "PDN-1146 - Kafka SSL beendet: 28 Sek." (PDN-1146 - Kafka SSL finished: 28 seconds), with 42 replies and a meeting duration of 2 minutes and 52 seconds.
- A blue banner for a meeting titled "weitermachen mit durchstich" (continue with through-strike) at 01:02:23, with a "Teilnehmen" (Join) button.
- A message from "PD" stating "durchstich arteria->cp->xpdp->pdn testen gestartet" (through-strike arteria->cp->xpdp->pdn testing started), with 8 replies.
- A blue banner for a meeting titled "PRMRY-1987" at 13:13, with a "Teilnehmen" (Join) button.
- A message from "PD" stating "PRMRY-1987 gestartet" (PRMRY-1987 started), with replies.

At the bottom of the chat, there is a button labeled "Neue Unterhaltung" (New conversation). The left sidebar shows the "T - Mob Sessions" team selected, and the bottom left corner has a "Teams verwalten" (Manage Teams) button.

Egal ob remote oder vor Ort ...

Es wird 2er-„Mobs“ geben ...

Euer Pair Programming wird besser

durch das (Aus-)Üben von Mob Programming!

"Pair Programming is like going on a date.
Mobbing is like having dinner with friends.
Different dynamic."



Allen Holub
@allenholub

Antwort an @BillyGarnet und @mcflyDev

I see Mob Programming as an improvement over Pair Programming. They're both useful, of course, but I've seen Mob Programming be successful in places where Pairing didn't work. Pair Programming is like going on a date. Mobbing is like having dinner with friends. Different dynamic.

6:34 nachm. · 11. Mai 2020 · TweetDeck

„Driver“ oder „Typist“?

Pair Programming – Driver sein kann stressig sein

1:1

Zu großer Unterschied in Wissen & Können?

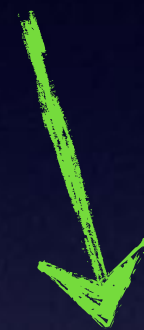
Mob Programming – Entspannen an der Tastatur

Abwarten, zurücklehnen & die anderen denken lassen

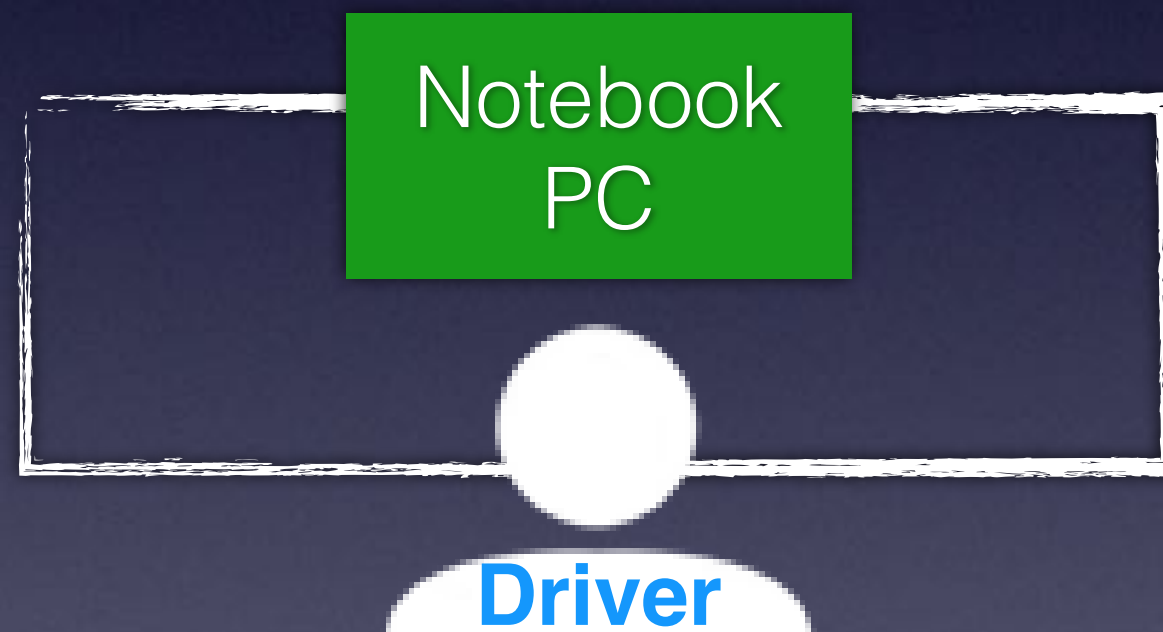
(Quelltext-)Protokoll mitschreiben

Mob Testing

Anwendung
(App)
bedienen



Großer Bildschirm (Beamer)



Exploratives Testen
etc.



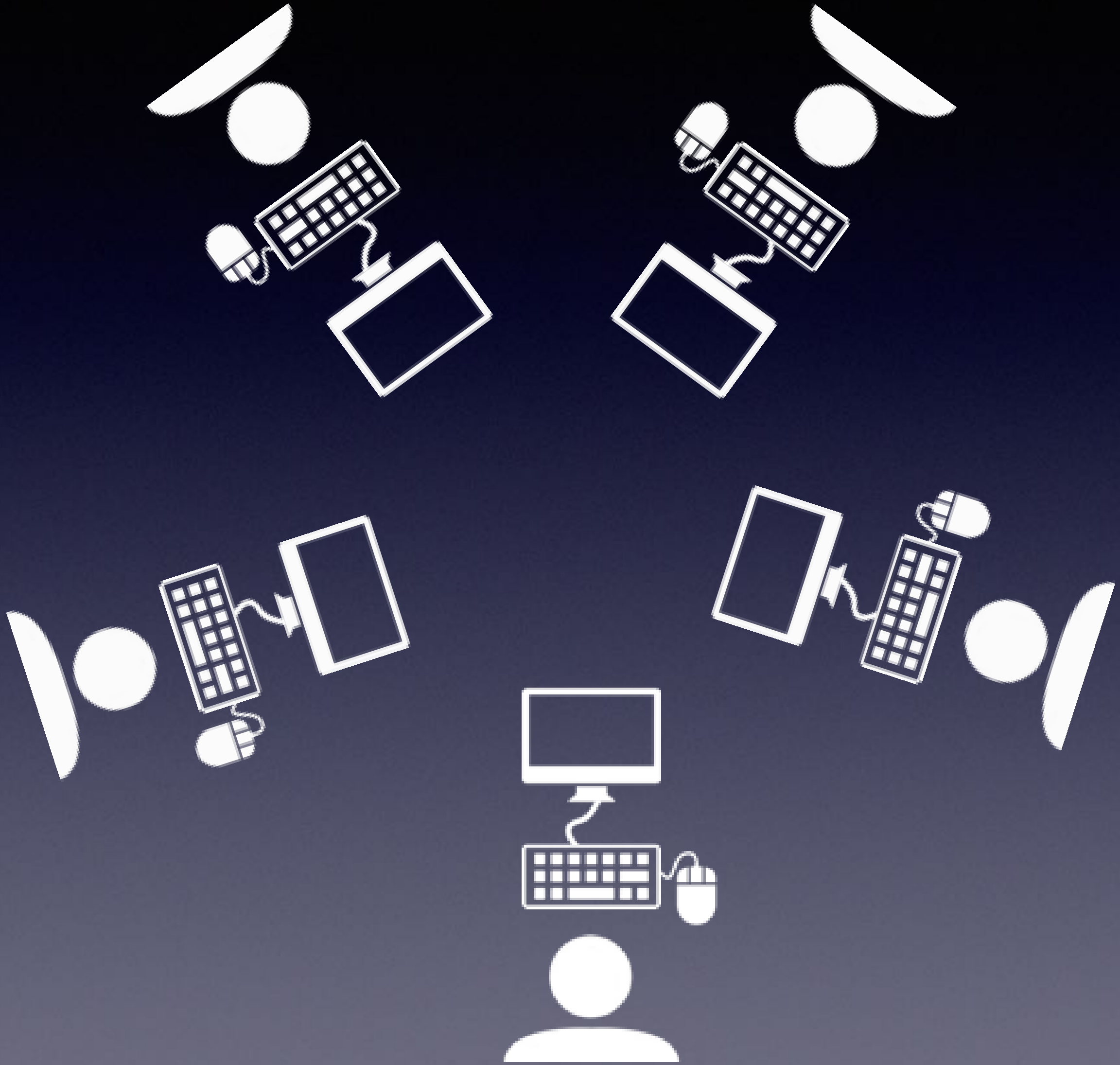
Testprotokoll
etc.



Warum funktioniert
(Remote) Mob Programming?

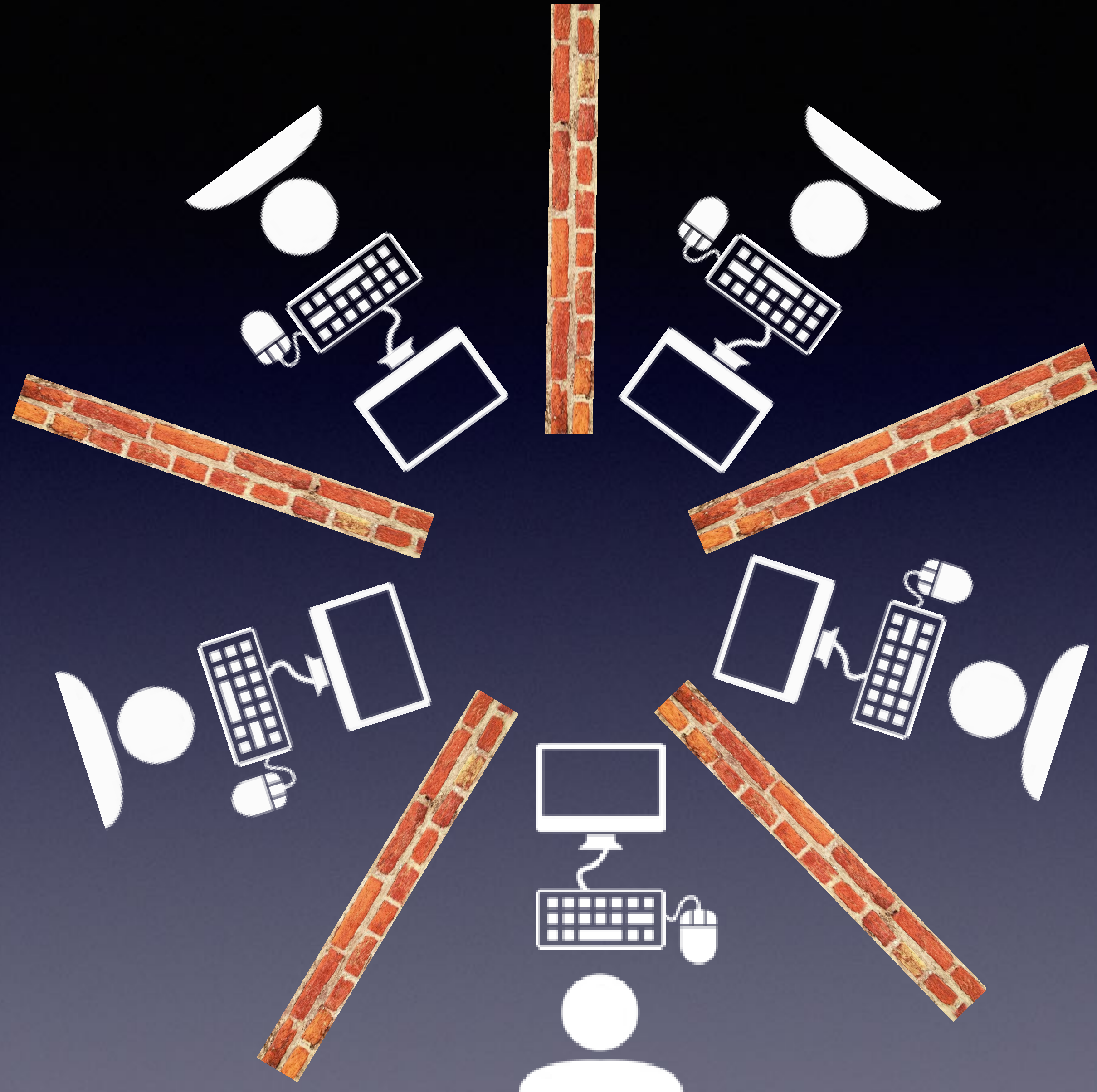


In Eurem Büro



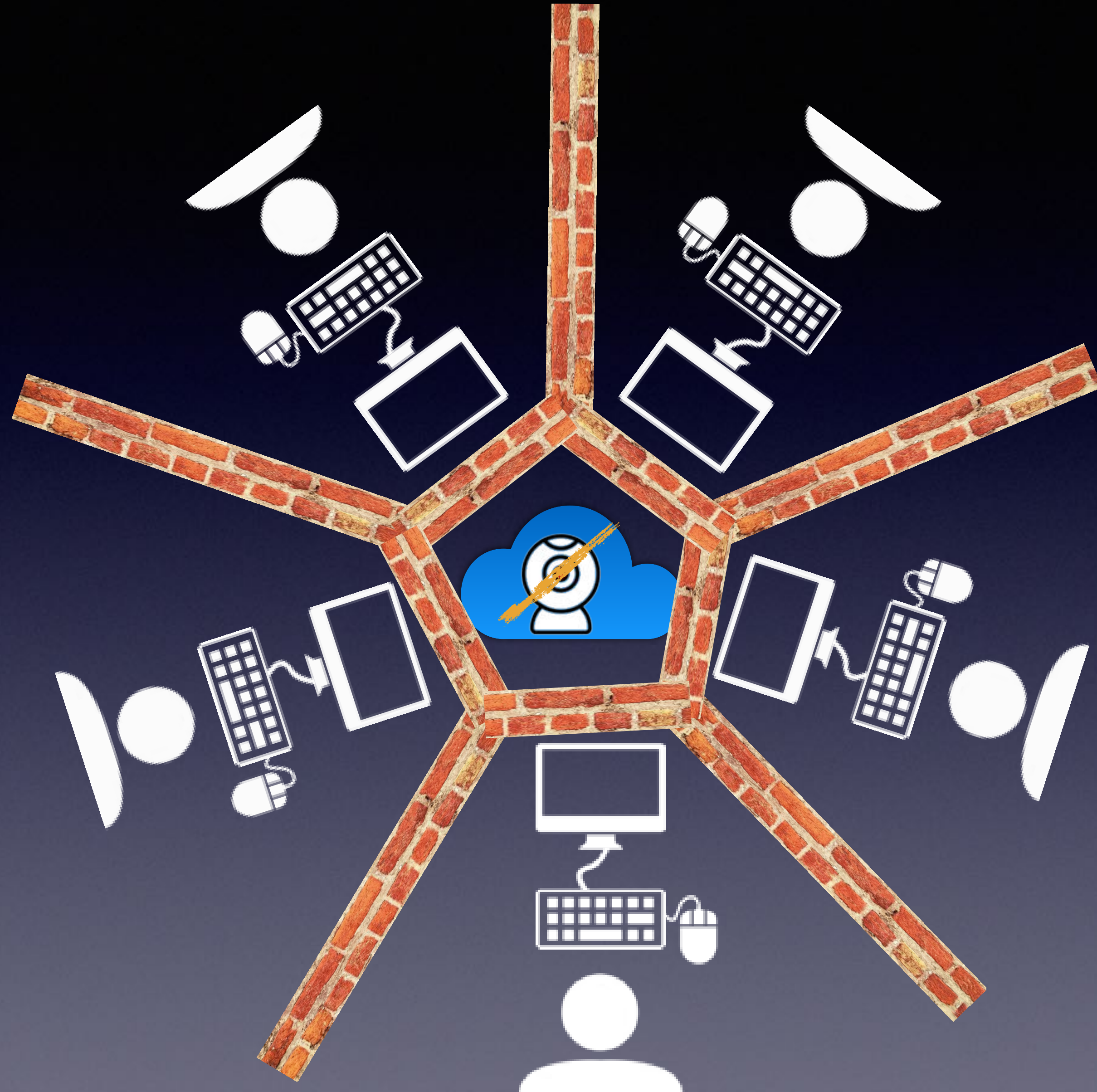
Ein Team?

In Eurem Büro



"Teams" Einzelner?

Home-office



Abtauchen
hinter dem
Bildschirm...

Remote? Nicht das Problem.

Anwesenheit vor Ort *keine Voraussetzung* für Team-Zusammenhalt!

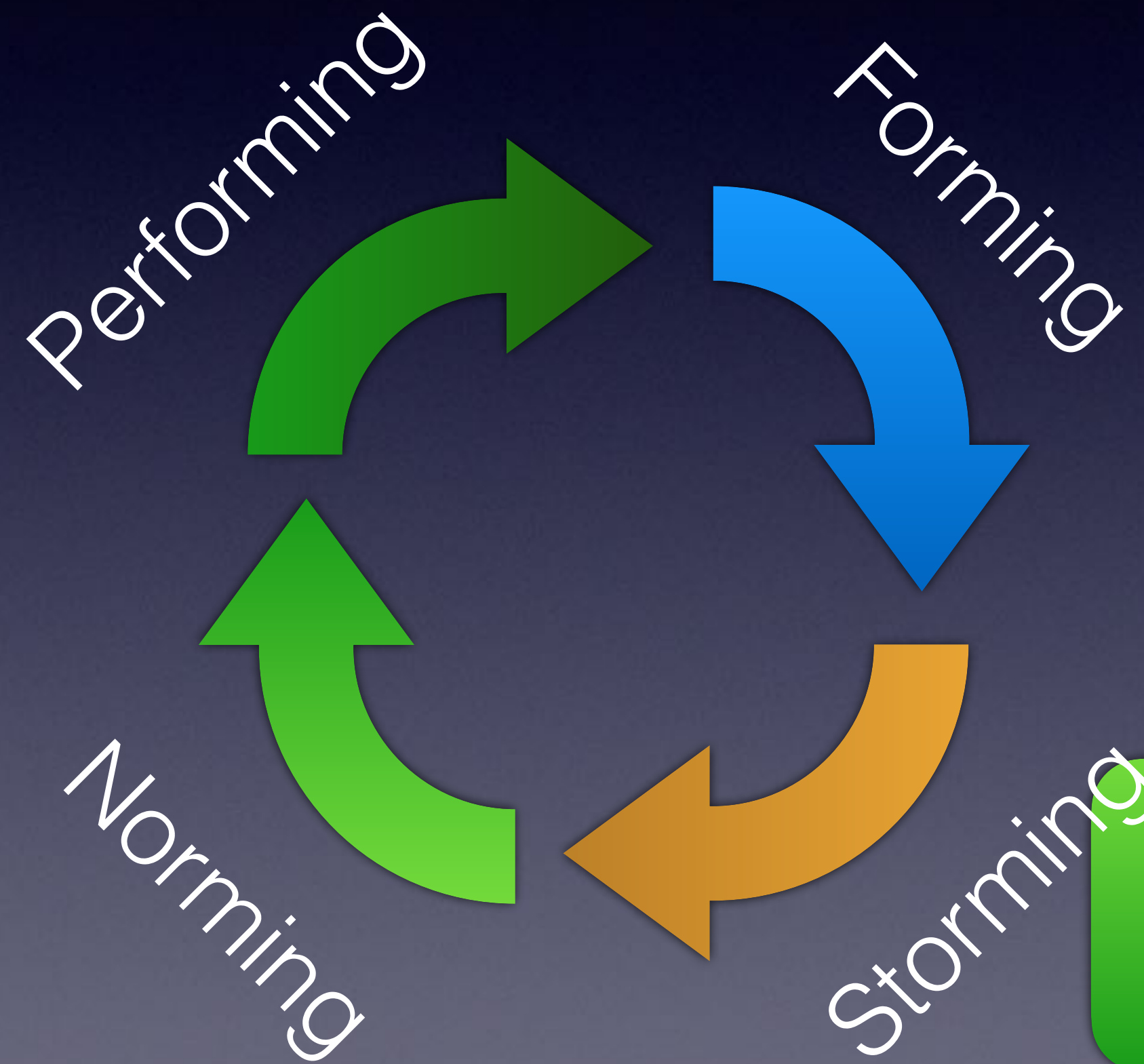
Wir brauchen:

Gemeinsame Werte

Vertrauen

Die anderen regelmäßig sehen

Phasen der Gruppenentwicklung



Kann man nicht.

Oft vermieden...

Konflikte müssen
geklärt werden

Phasen der Gruppenentwicklung



Stabile Teams?

Teams sind **unveränderlich**

Veränderung passiert

Neues Team, neues Forming/Storming/...

Mob Programming

Kürzere Phasen,
weil regelmäßig
durchlaufen!



Optimieren für Veränderung

~~Widerstand gegen Veränderung?~~

Widerstandsfähigkeit*
gegenüber Veränderungen!

*) Resilienz

Wissensverteilung?

Dokumentation.

Nachträglich.

"Show & tell"

Erstaunlich
schwierig...

Mob Programming

macht implizites (stilles)

Wissen & Können

sichtbar

Anderen beim Denken zuschauen.

Beobachten. Erleben. Selber machen.

Jemand anderes
fragt „meine“ Frage
(die ich nicht
formuliert bekomme)

Wissen **und Können**

Know how

Collaborate?

Co-**Author**, Co-Create!

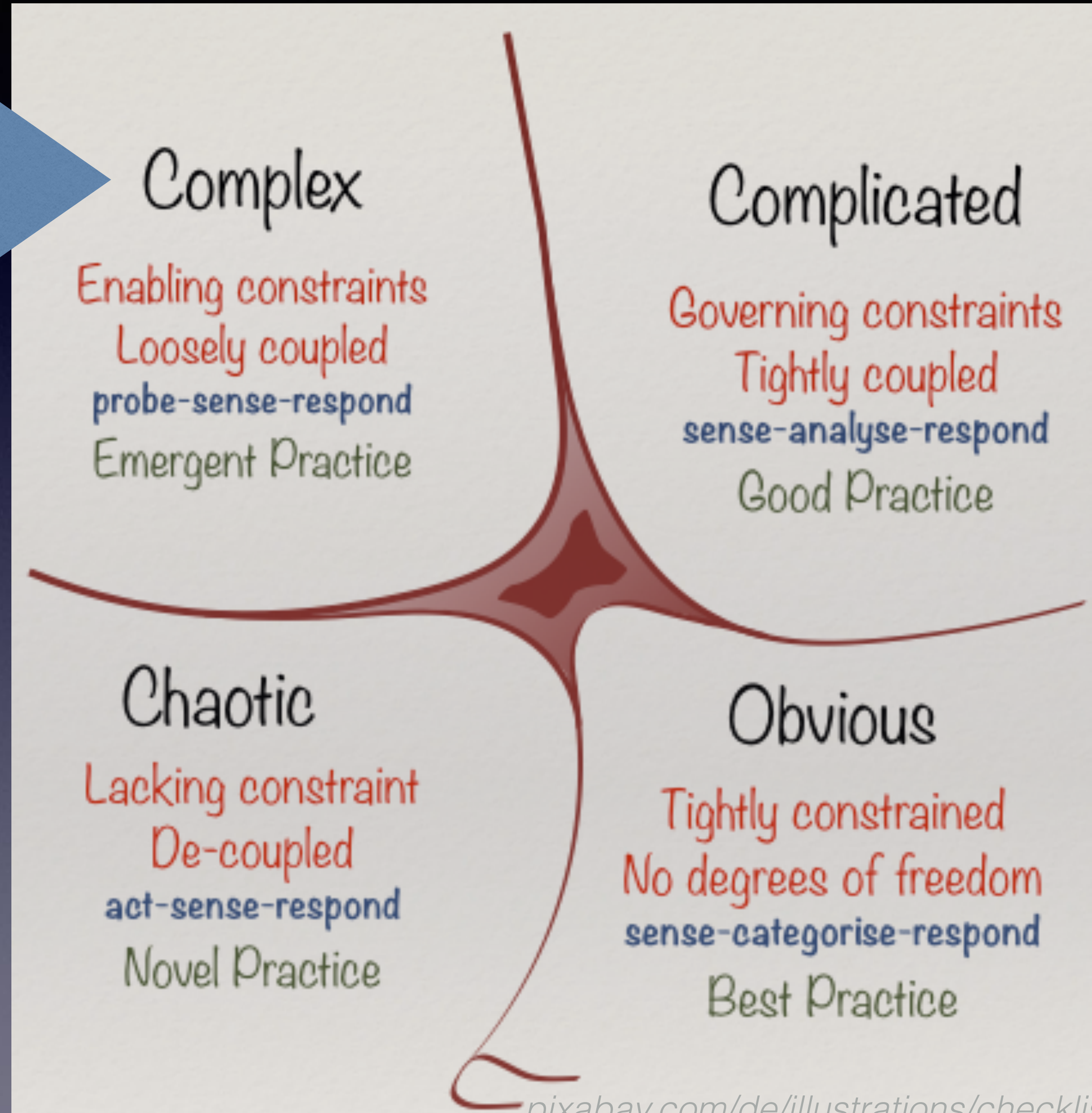
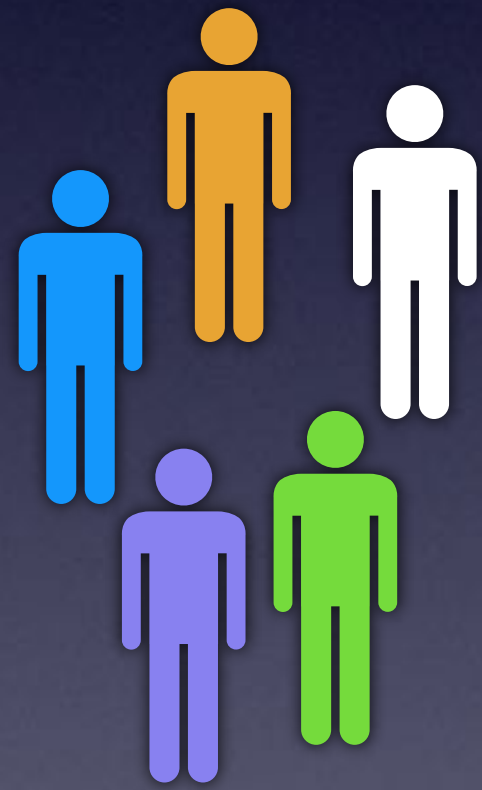
Team muss den Lösungsw**eg** verstehen

(nicht nur das Ergebnis)

„Warum ist Code *nicht* da?“

Cynefin

Wir sind oft hier



The Cynefin Framework
by Dave Snowden.
CC BY-SA 3.0

3 Stufen von Mob Programming

Verschiedene
Team-Phasen

Gemeinsam lernen

Kurzfristig: Wissen &
Können weitergeben

Gemeinsam produktiv sein

Mittelfristig:
Wartbarkeit &
Collective Code
(Product) Ownership

Gemeinsam ausprobieren

Langfristig:
Kulturwandel

Verschiedene
Problem-Kategorien

100% Mob Programming?

Geht! Aber definitiv kein Muss.

(wie beim Pair Programming)

Hilfsmittel, Werkzeug

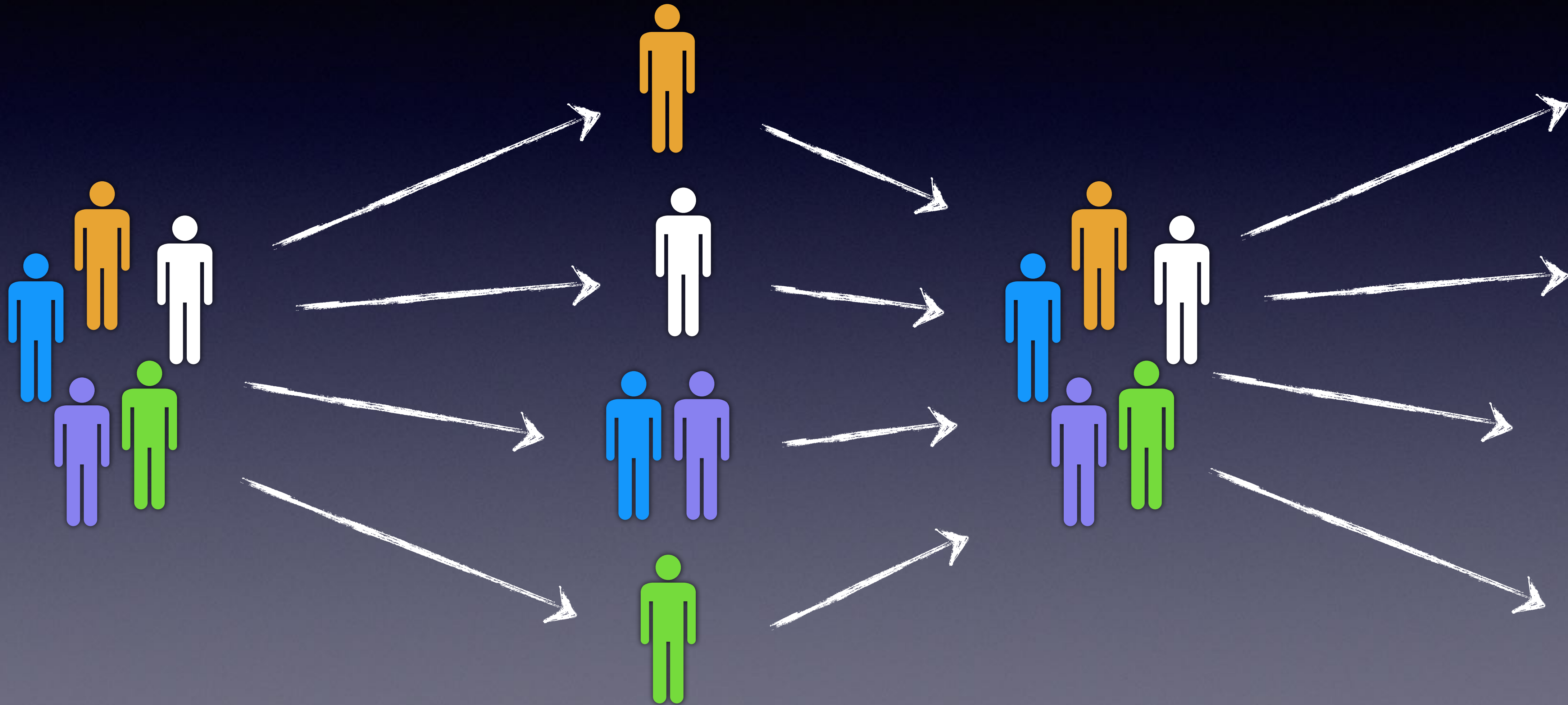
100% für die wichtigste Aufgabe

... aber nicht zwingend 100% Anwesenheit jedes Einzelnen!

Dynamischer Mob: Kommen und gehen

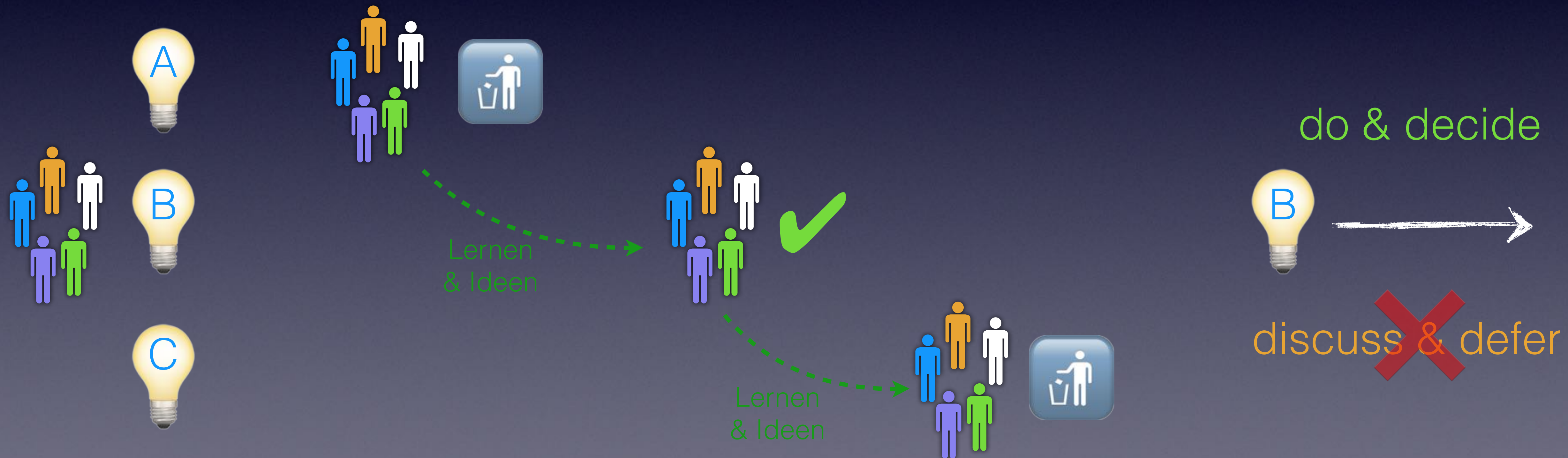


Zusammen starten!



Team-Entscheidungen

Mehr als eine Idee? *Probiert alle aus!*



Anders als Meetings!

Am Ende haben wir

lauffähigen, getesteten, **verstandenen Code**.

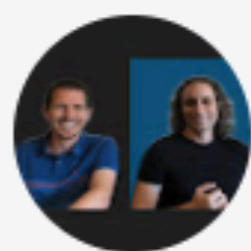
Das **motiviert** uns Entwickler:innen 😊
& macht unsere Kund:innen **glücklich** 😊

“If you want to go **fast**, go **alone**.

If you want to go **far**, go **together**.”

– Sprichwort

MOB MENTALITY



Mob Mentality Show

755 subscribers

SUBSCRIBE

HOME

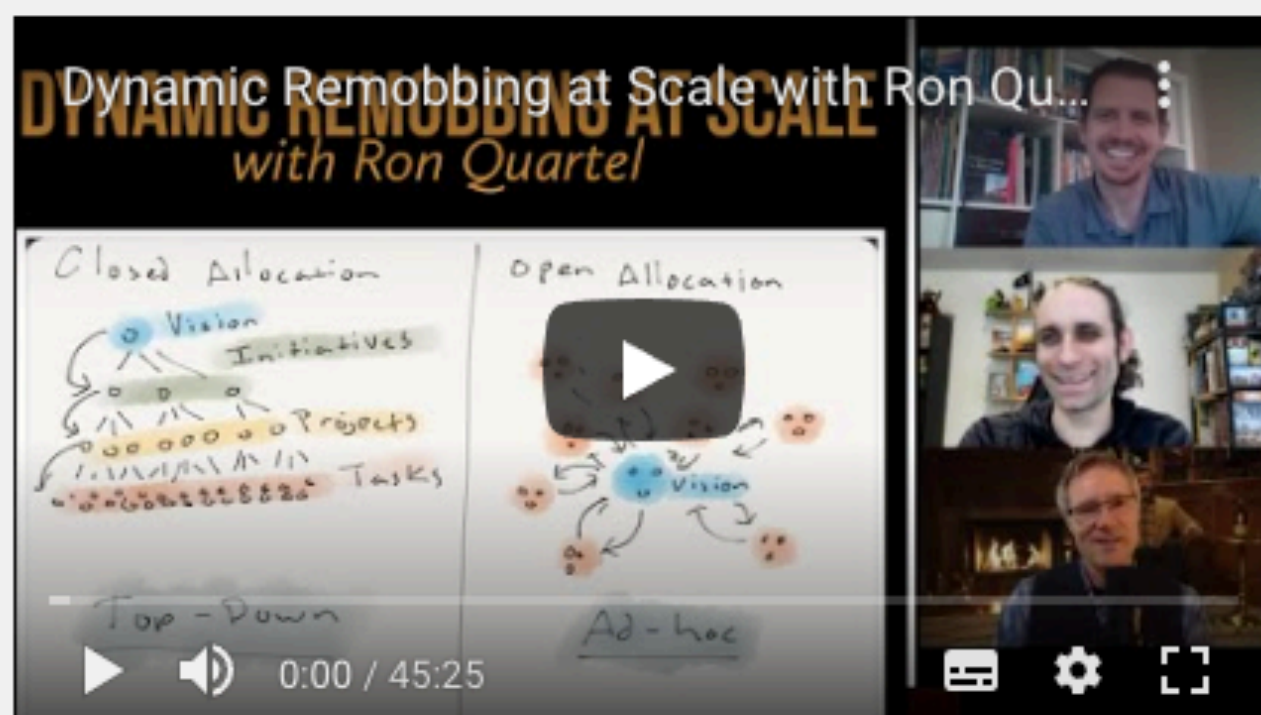
VIDEOS

PLAYLISTS

CHANNELS

DISCUSSION

ABOUT



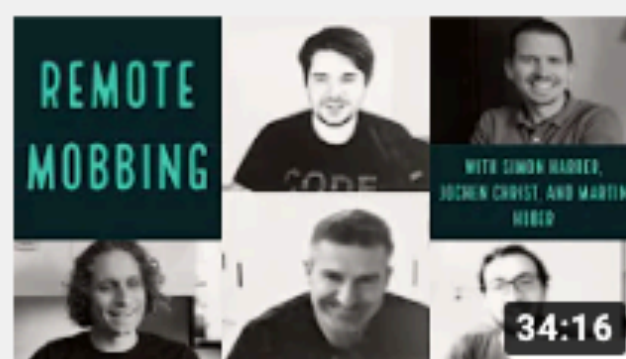
Dynamic Remobbing at Scale with Ron Quartel

34 views • 18 hours ago

Chris Lucian and Austin Chadwick discuss all things #agile and product development from a #MobProgramming perspective. Imagine a team of 30-150 people that turned up the dial on self-organization to the max. Instead of favoring top-down closed allocation, they embraced open-allocation and autonomy. Now also imagine that many of these people happen to love #PairProgramming and #MobProgramming. What would their

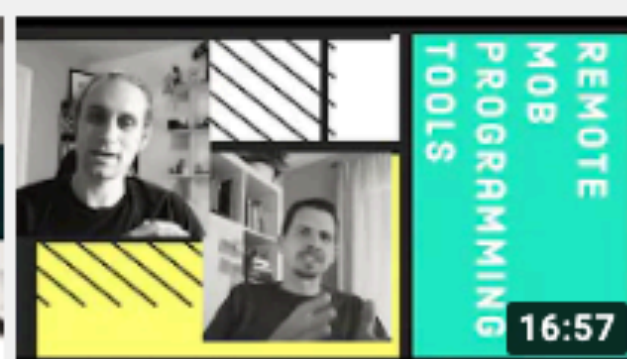
READ MORE

Remote Mob Programming ▶ PLAY ALL



Remote Mobbing with Simon Harrer, Jochen Christ, and...

Mob Mentality Show



Remote Mob Programming Tools

Mob Mentality Show



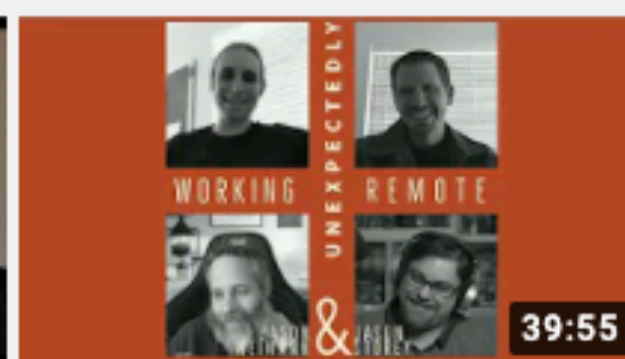
Remote Mob Programming Etiquette

Mob Mentality Show



Remote Collaborative Coding with Lynn Langit

Mob Mentality Show



Unexpectedly Working Remote with Jason Weiman...

Mob Mentality Show

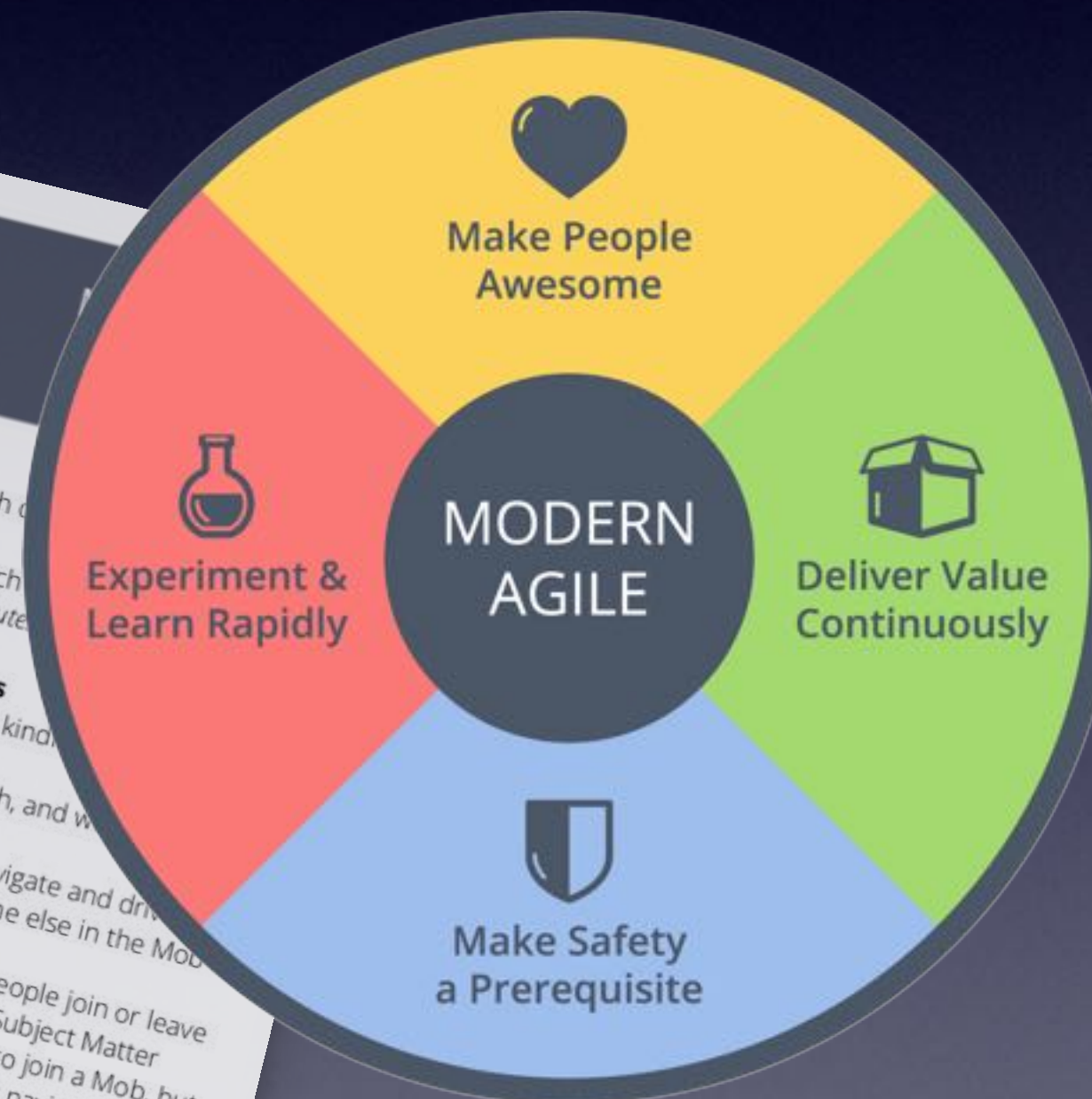
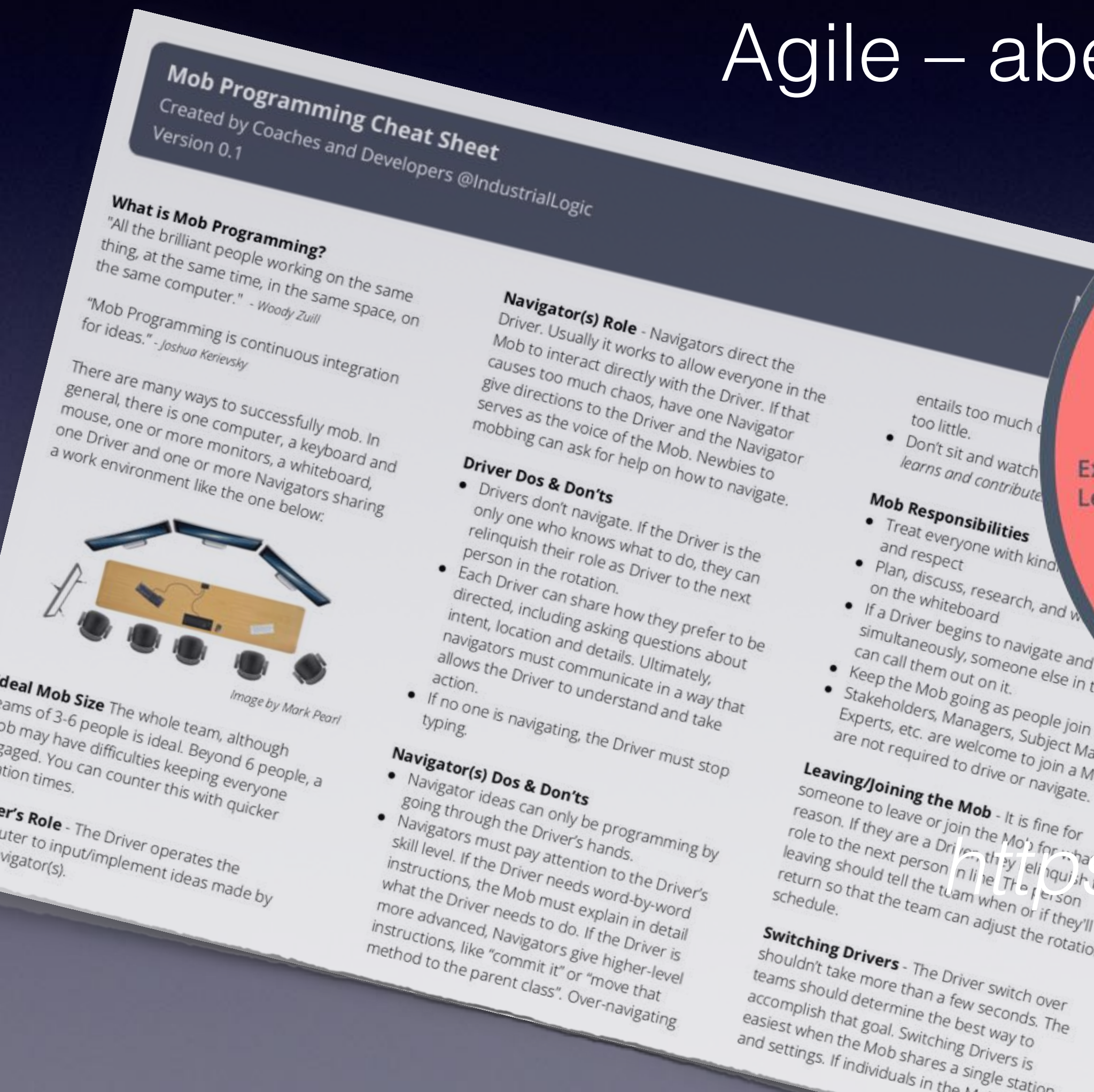


A Remote UI/UX Mobbing Story with Damien Klinnert

Mob Mentality Show

Modern Agile

Agile – aber so, dass es funktioniert.



#MobProgramming

#NoEstimates

#NoProjects

#ModernAgile

<https://modernagile.org/>

Mob Programming



Know-How-Transfer

Pausen machen

Lesbarkeit

Einfachheit

Geschwindigkeit

Collective Product Ownership

Fragen?

Pair

Programming

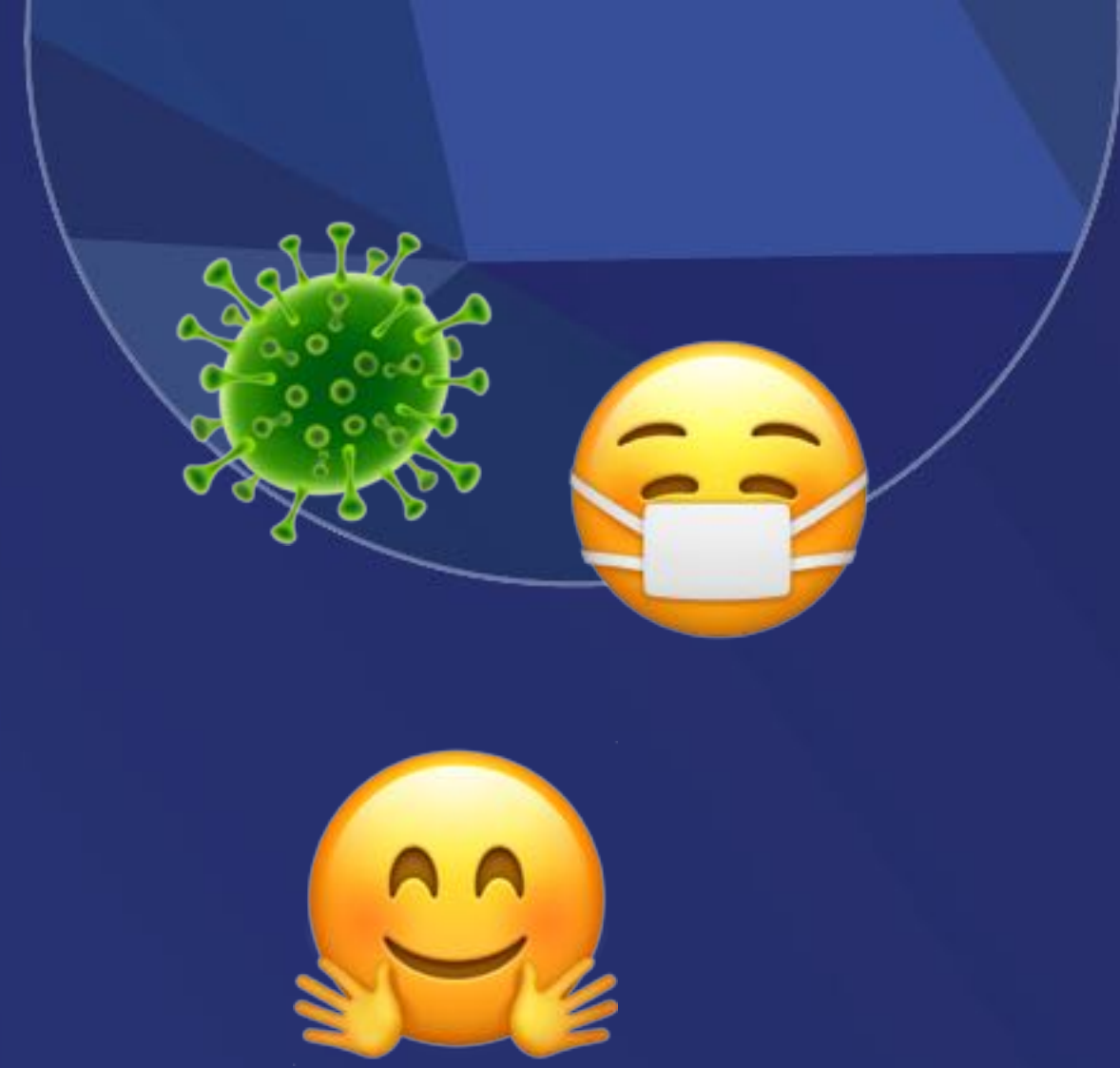
Coaching

XP Remote

Modern Agile

Strong Style Navigation

TDD



Vielen Dank!

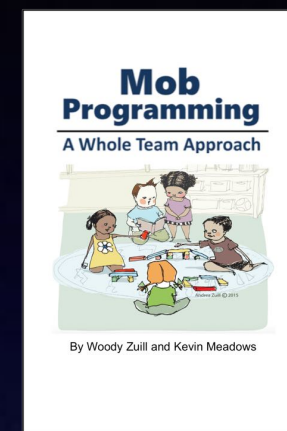
"We are uncovering better ways of developing software by doing it and helping others do it." *

thomas@muchsoft.com

 @thmuch

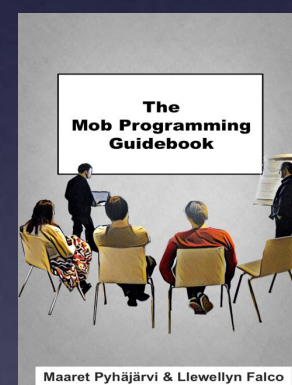
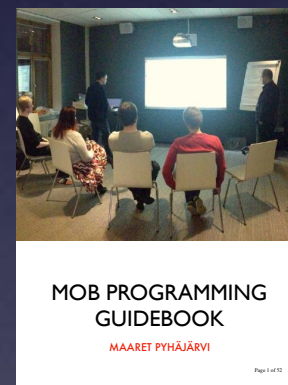
*) <https://agilemanifesto.org/>

Weitere Informationen



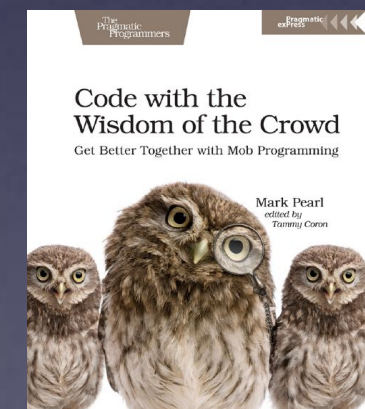
<https://mobprogramming.org/>
<https://leanpub.com/mobprogramming>

<https://www.remotemobprogramming.org/>
<https://leanpub.com/remotemobprogramming>

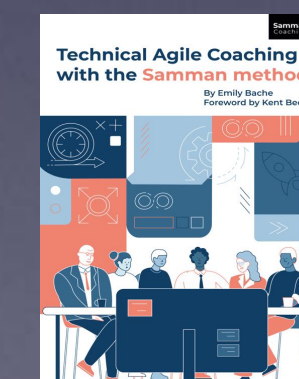


<https://mobprogrammingguidebook.xyz/>
<http://www.mobprogrammingguidebook.com/>

<https://pragprog.com/book/mpmob/code-with-the-wisdom-of-the-crowd>



<https://leanpub.com/techagilecoach>



<https://www.chrislucian.com/p/companies-that-are-mob-programming.html>

Code-Sharing

Git-Handover

<https://mob.sh/>

IntelliJ Plug-in: <https://plugins.jetbrains.com/plugin/14266-mob>

IDE-Sharing

<https://www.jetbrains.com/de-de/code-with-me/>

<https://visualstudio.microsoft.com/services/live-share/>

<https://www.codetogether.com/>

<https://gitduck.com/>

<https://git.live/>

<https://codeshare.io/>

<https://gitpod.io/>

<https://theia-ide.org/>

<https://aws.amazon.com/de/cloud9/>

Cloud-Dev-VM

<https://claremacrae.co.uk/blog/2020/12/virtual-machines-for-online-mobbing-and-training>

Online-Whiteboards

Miro

<https://miro.com/>

Mural

<https://www.mural.co/>

diagrams.net / draw.io

excalidraw.com

Mob-Timer

<https://cuckoo.team/>

<https://mobti.me/>

<https://agility.jahed.dev/>

<http://mobtimer.zoeetrope.com/>

<https://mobtimer.com/>

<https://saschamz.github.io/mobtimer/>

<https://github.com/MobProgramming/MobTimer.Python>

<https://github.com/pluralsight/mob-timer>

<http://mobster.cc/>

Am Code herumprobieren

Als Mob **experimentieren**

Selbstverständlichkeiten erfragen

Angstfreie Kommunikation

*Schnell viel Code ist **nicht das Ziel!***

Entspannen an der Tastatur

Einfache, kleine Schritte anstreben

Komplexität kommt früh genug

Mob Programming

"It's about getting the **BEST** (not the **most**) from your team."

– Llewellyn Falco

"**All the brilliant minds** working on the same thing,
at the same time, in the same space, and at the same computer."

– Woody Zuill

"Continuous Integration of Ideas"

– Joshua Kerievsky