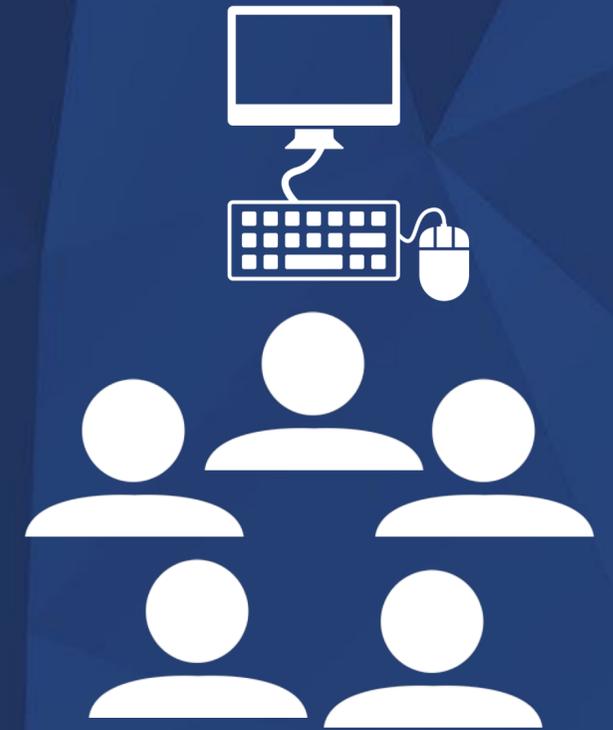




Mob Programming

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



Thomas Much

 @thmuch

07.05.2019

Softwareentwickler



Agile Developer Coach



 @thmuch



XP

Agile

Scrum

Software Crafting

Kanban

~ 20 Jahre alt

User Stories

Pair Programming

Refactoring TDD Iterationen

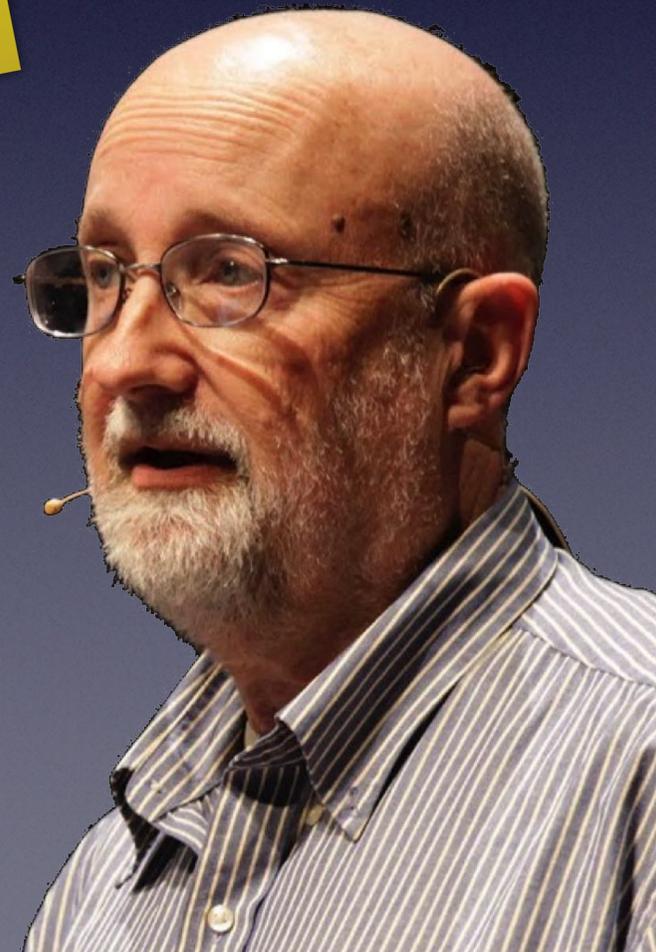
Continuous Integration

Mob Programming

~ 5 Jahre alt

(Idee schon älter)

Ziel heute:
Diese recht neue
agile Programmierpraktik
kennenlernen



Woody Zuill
2014

<https://woodyzuill.com/>

<https://twitter.com/woodyzuill>

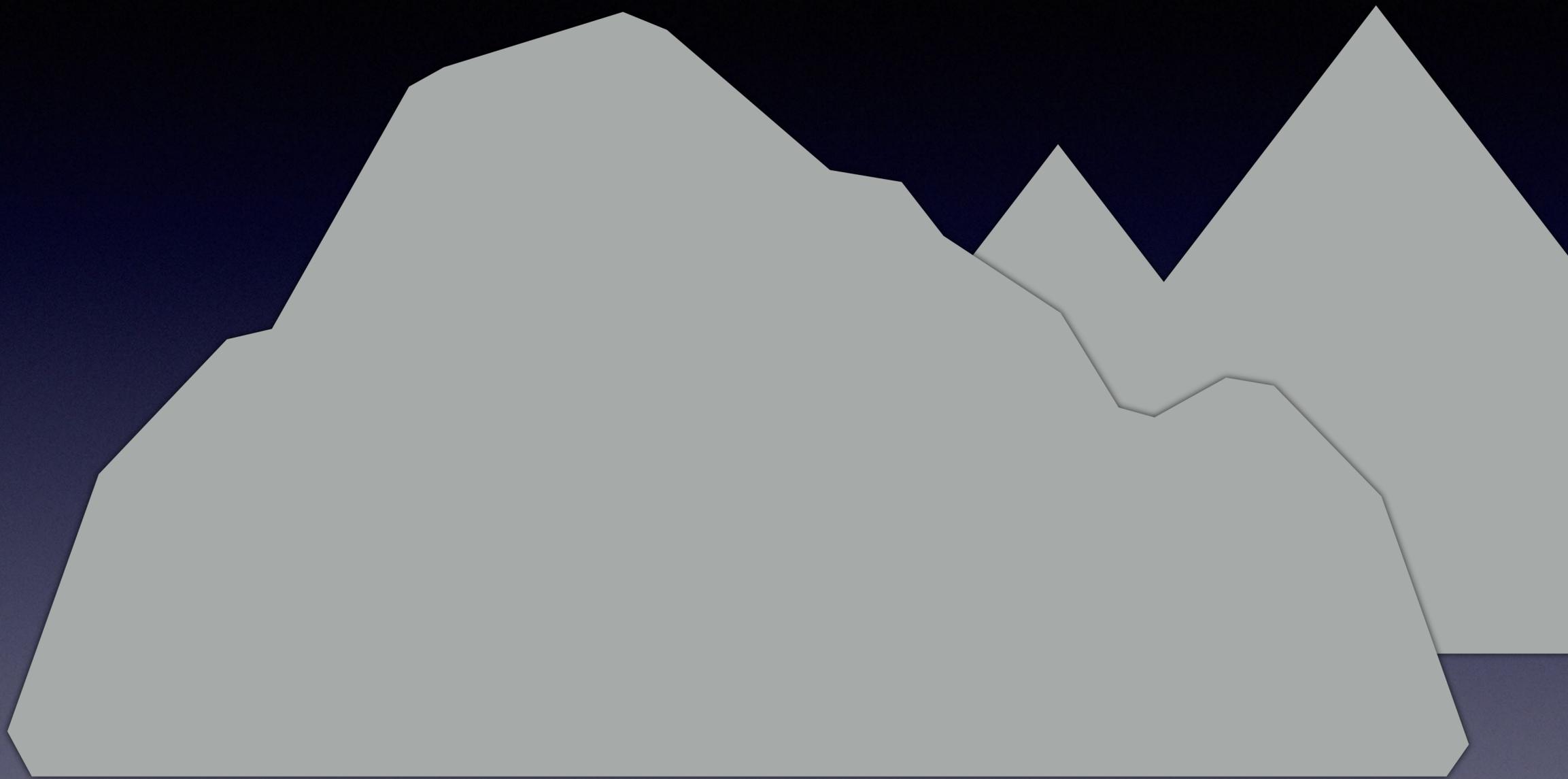
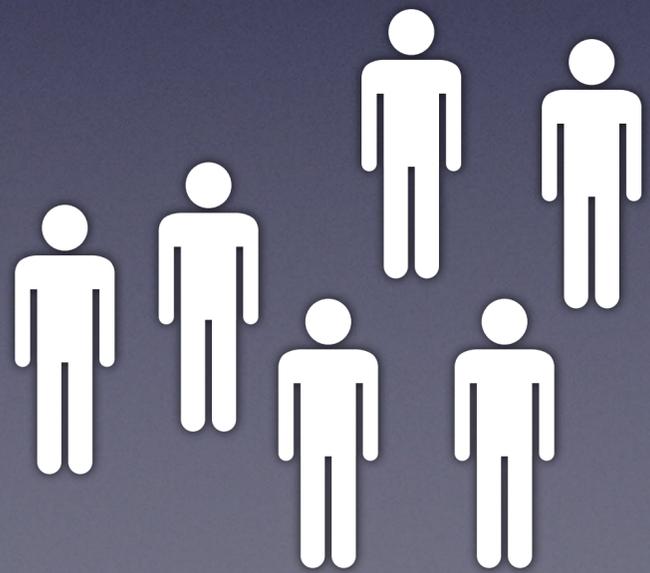
Mob Programming

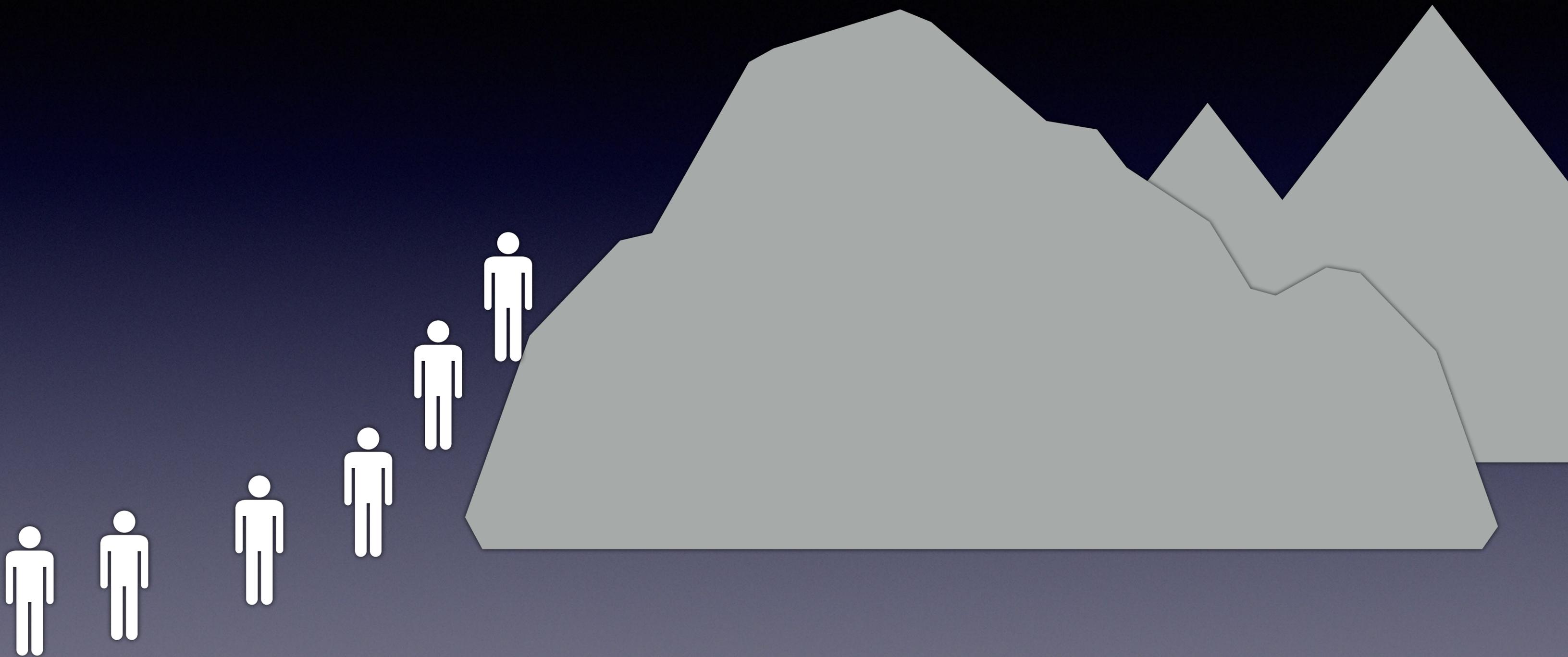
„Mobbing“?!

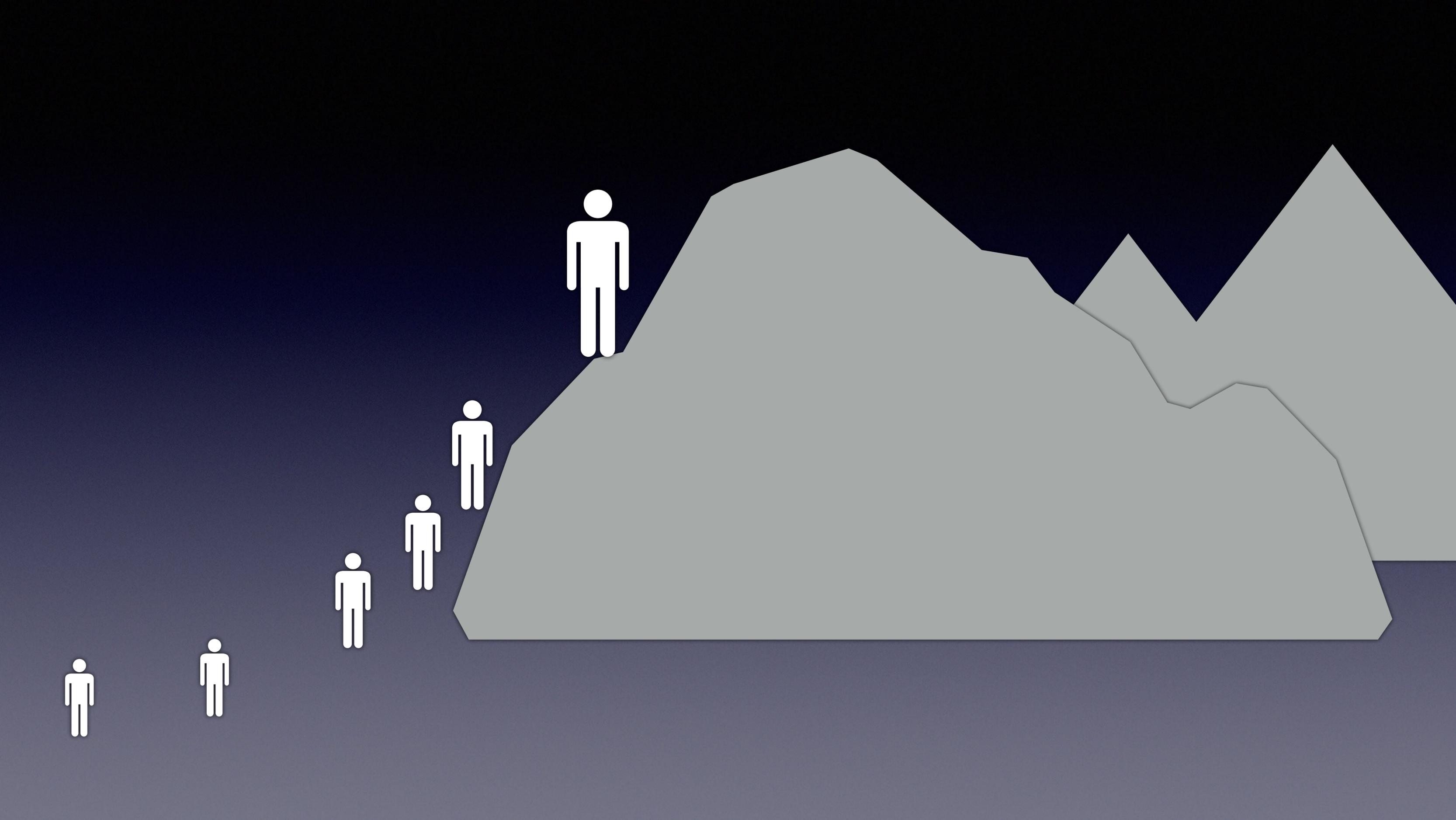
not „bullying“

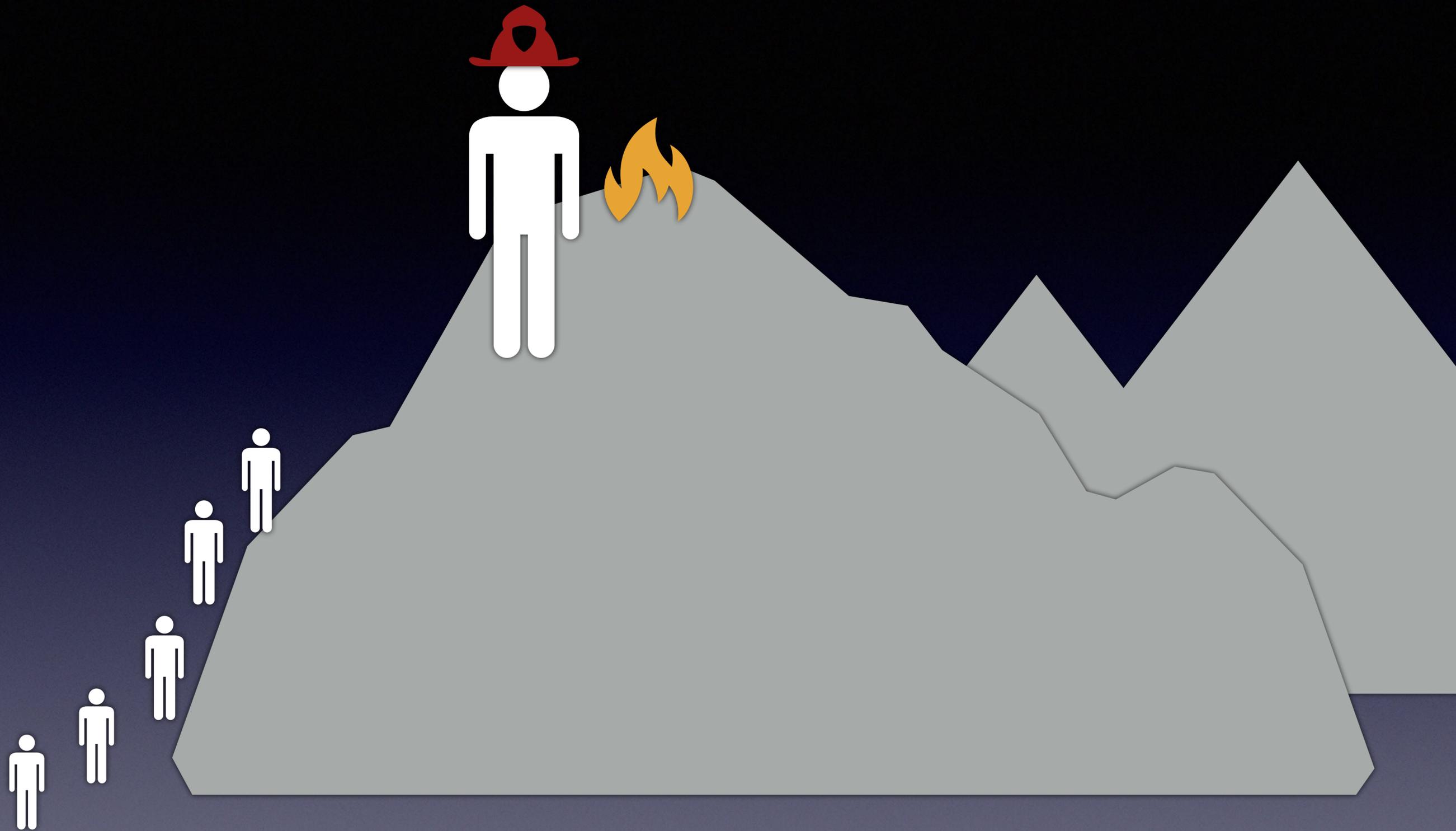
Group Programming

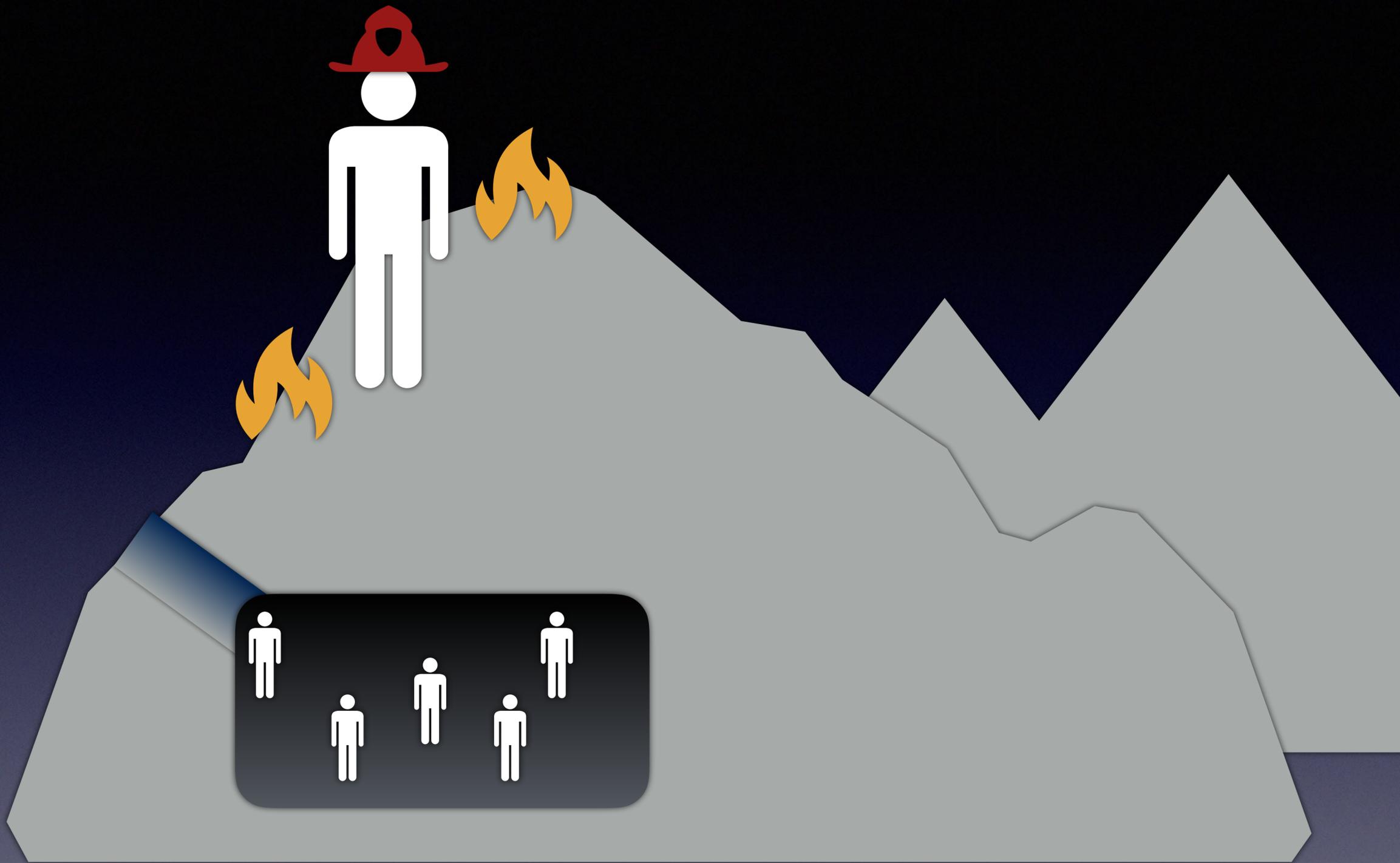
Wie arbeitet das Team?

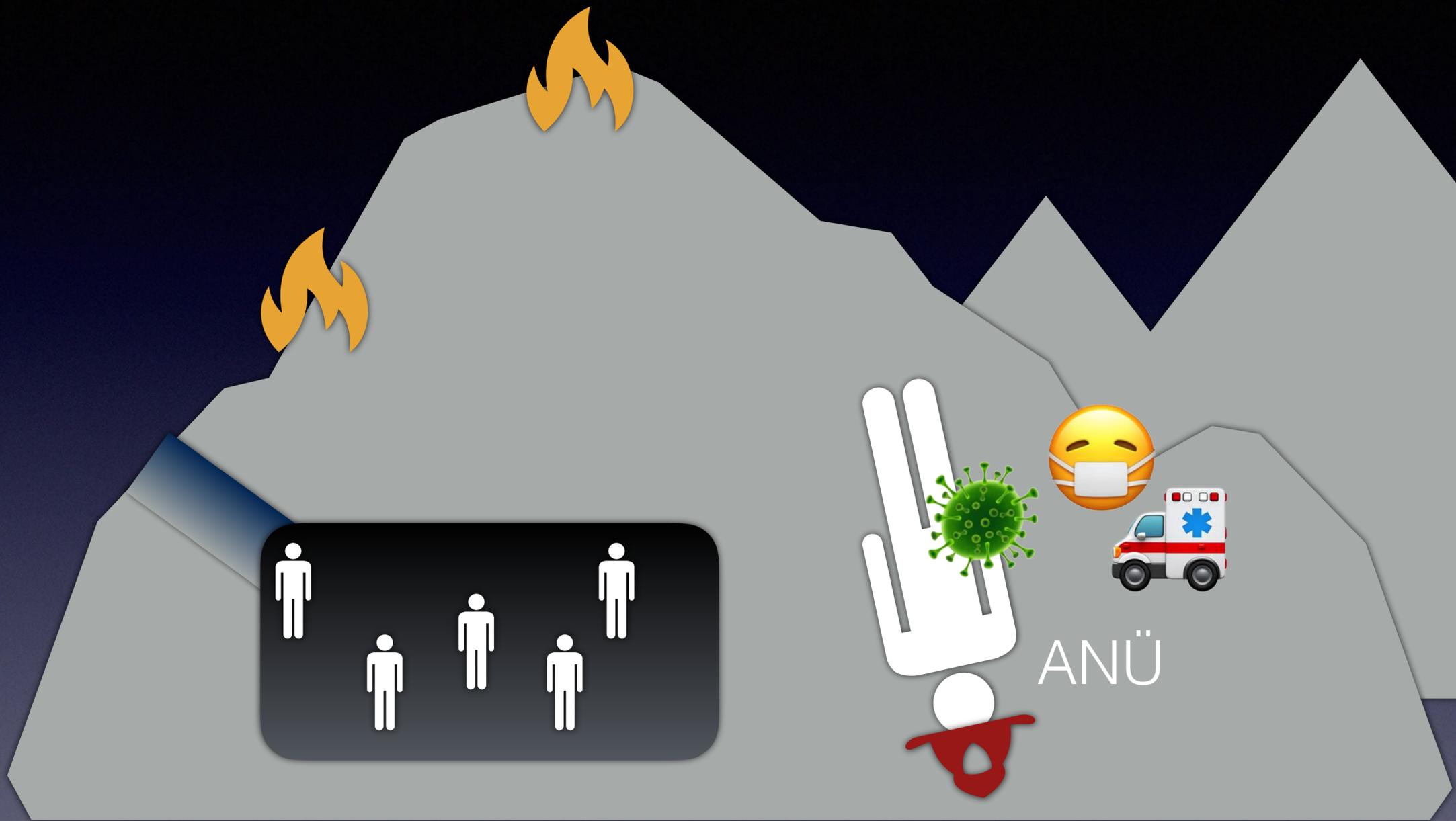




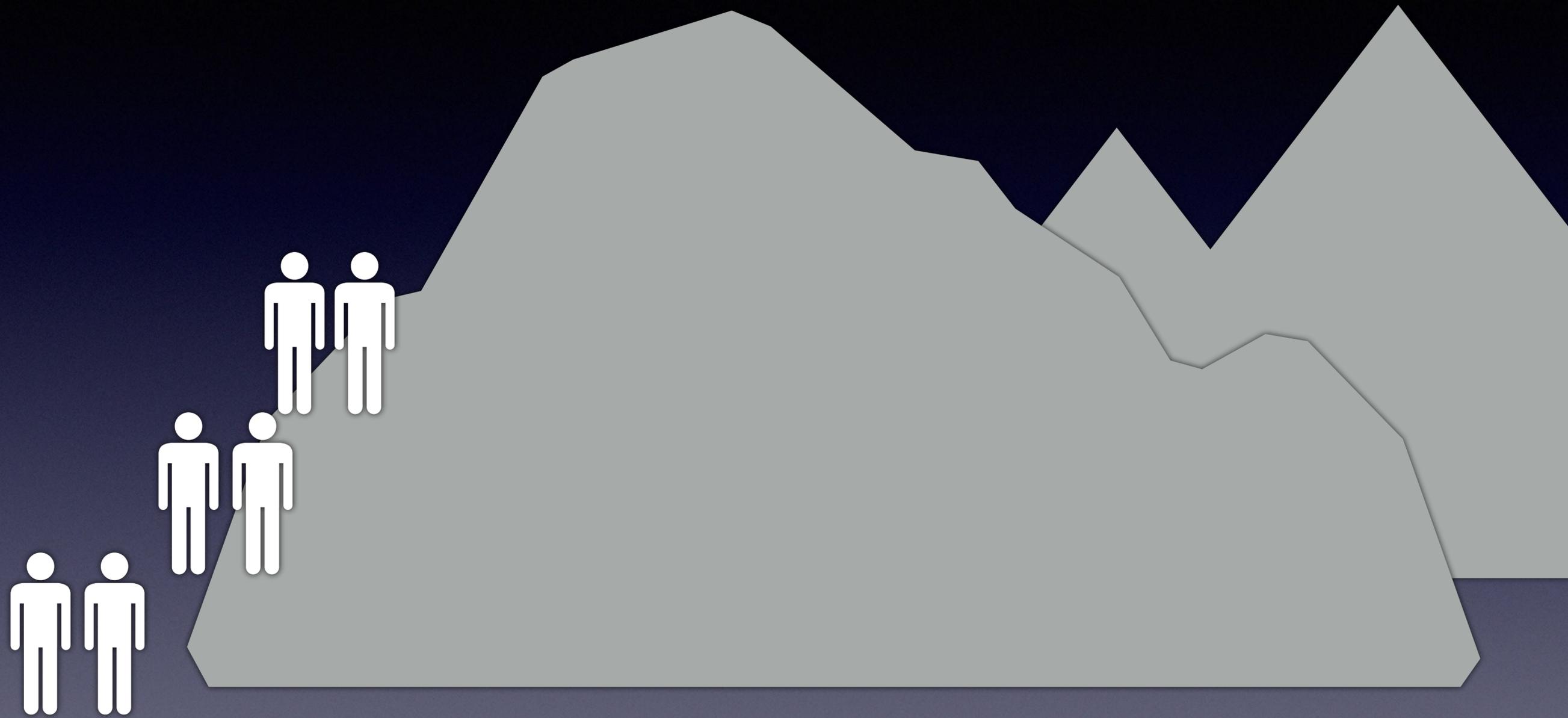




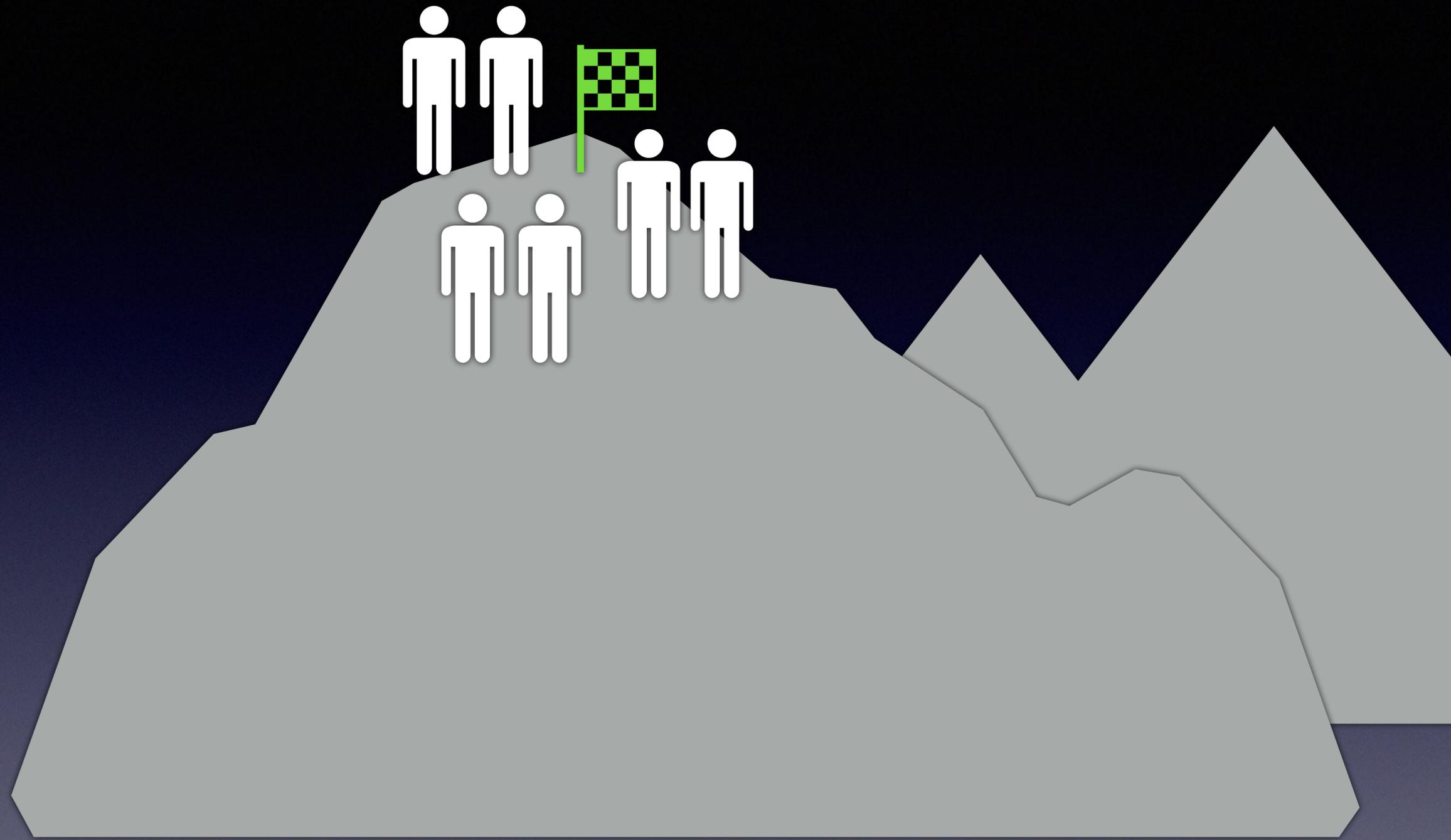




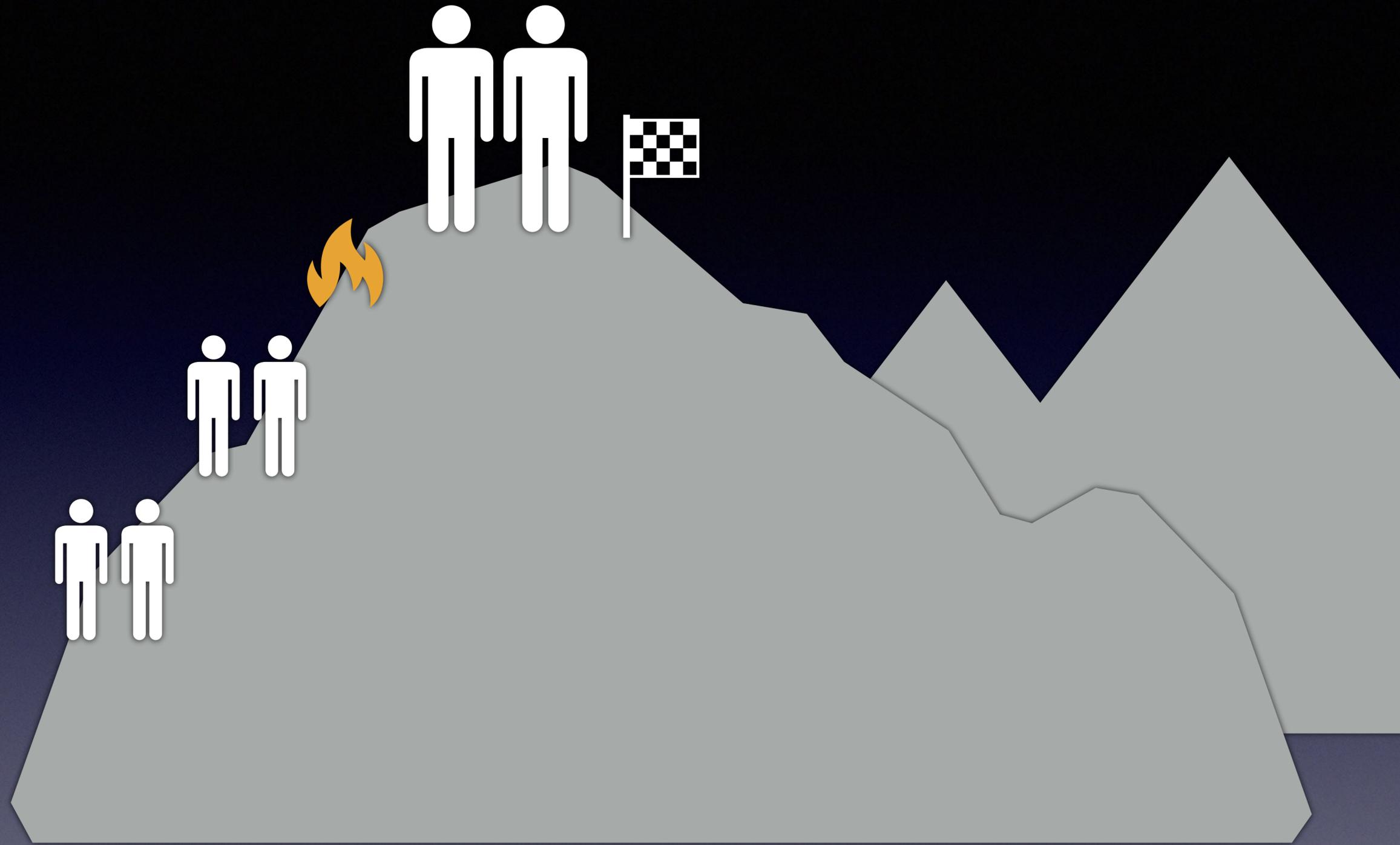
ANÜ



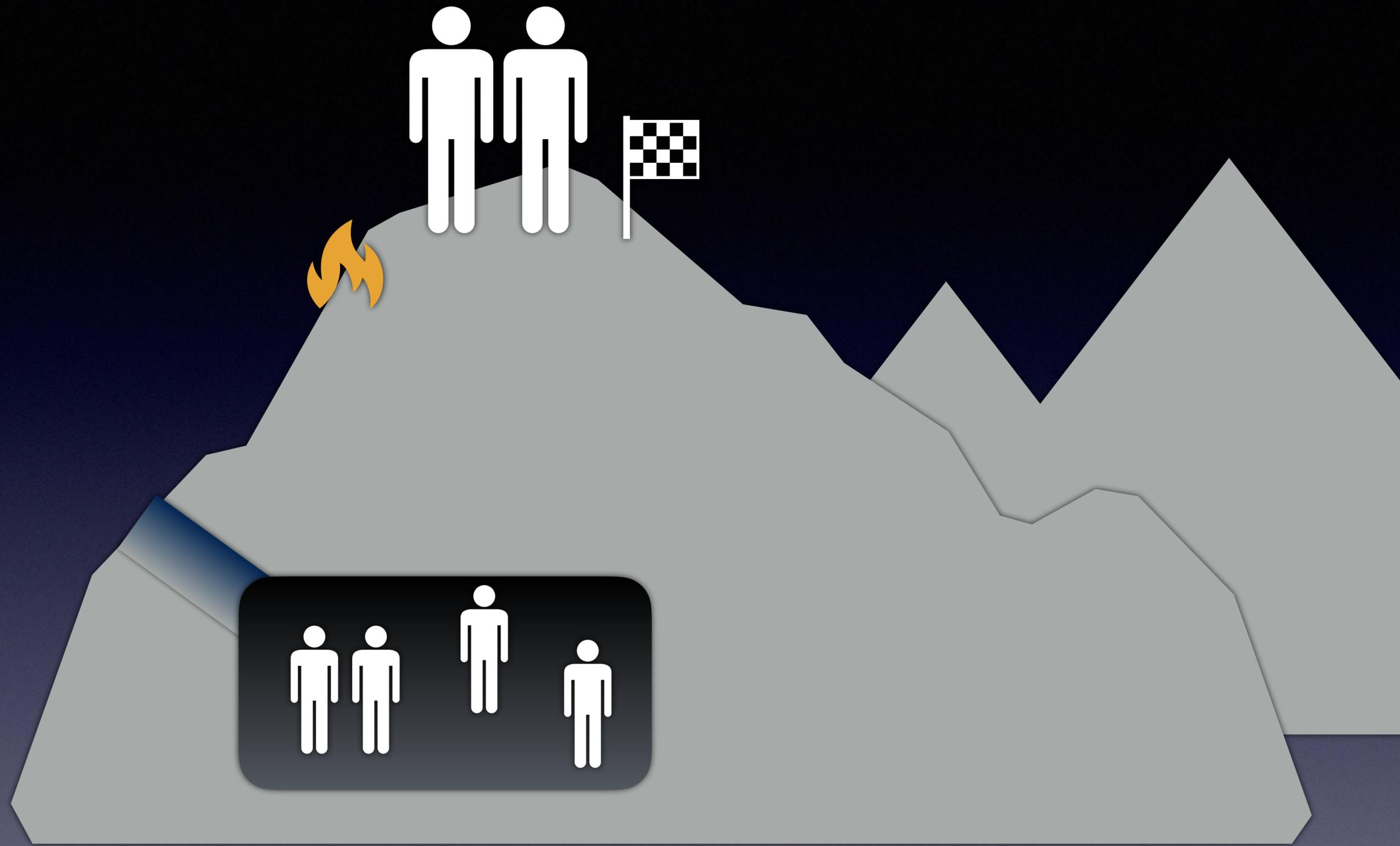
Pair Programming



Pair Programming



Pair Programming



Pair Programming

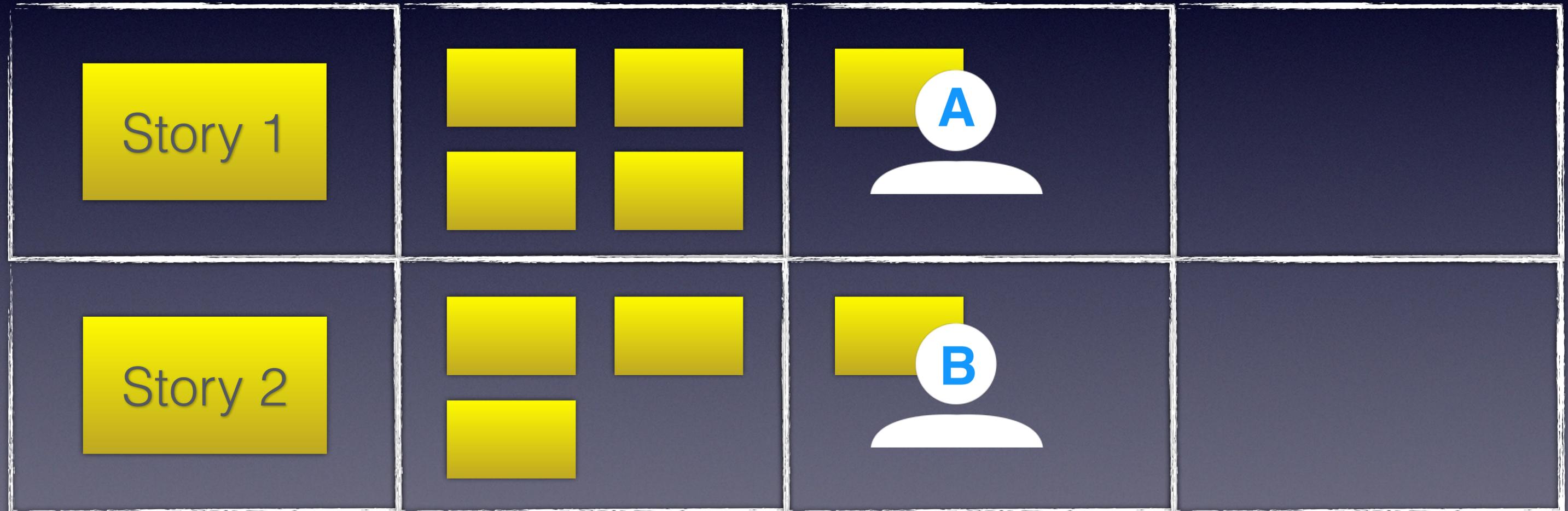
Wie arbeiten unsere Teams?

Arbeitet das Team agil?

To Do

In Progress

Done



„If your agile Team has individual work assignments,
I suspect it is neither agile nor team.“

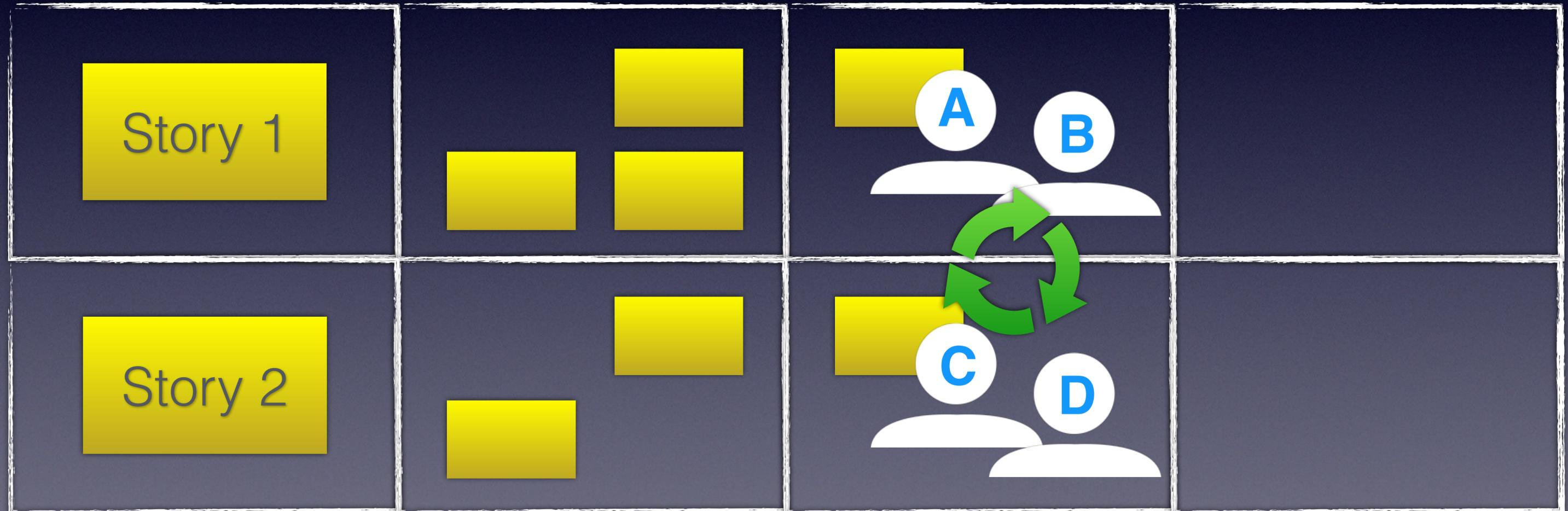
– **Tim Ottinger**

Echte Zusammenarbeit

To Do

In Progress

Done



Warum ist das wichtig?

Wissensverteilung

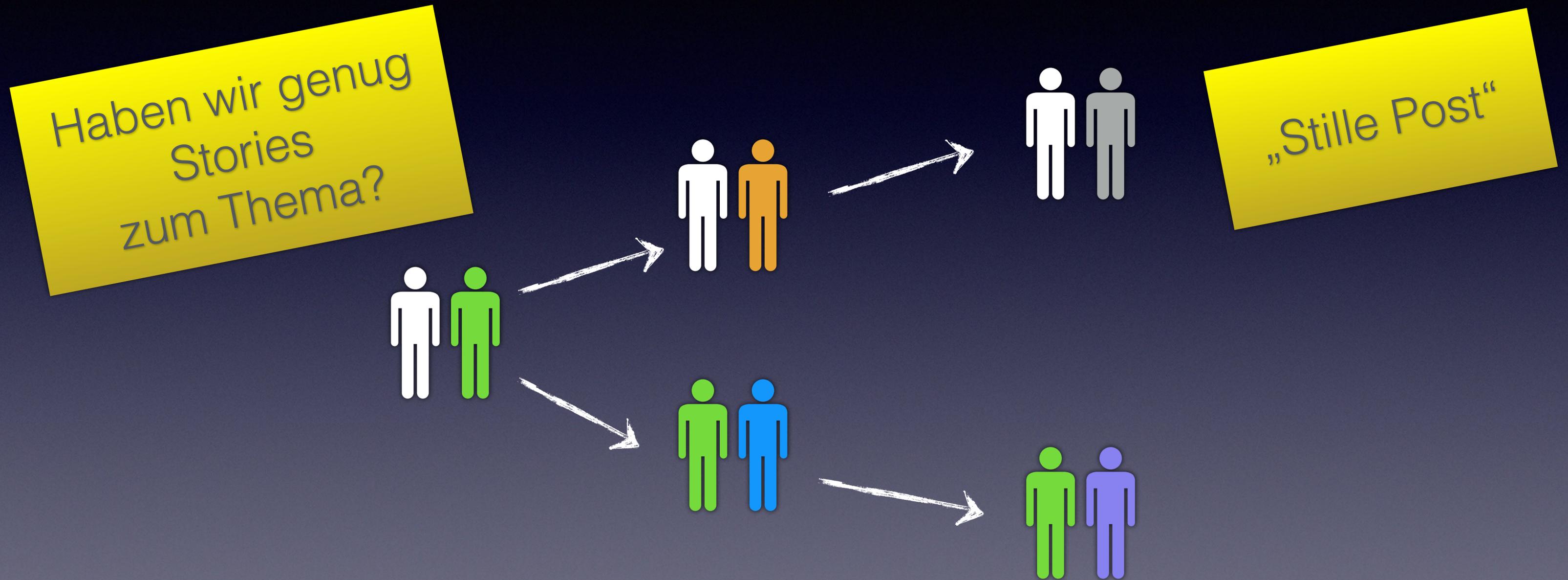
gemeinsame Konventionen

Collective Code Ownership

...

für Pflege / Wartung / Erweiterung / Anpassung der Software

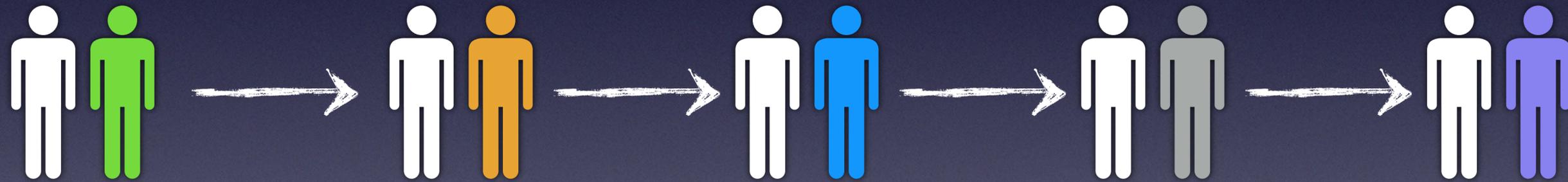
Wie verteilen wir Wissen etc.?



Wie verteilen wir Wissen etc.?

Haben wir genug
Stories
zum Thema?

„Einer
weiß alles“



1 Story
mit
diversen Tasks

Häufige Praxis:

Ein Pair erledigt die gesamte Story ...

Nachträgliche Wissensverteilung?

Oft unglaublich schwierig.

Warum diese Architektur?

Warum ist bestimmter Code *nicht* da?

Wie wäre es mit

gemeinsamem Wissensaufbau

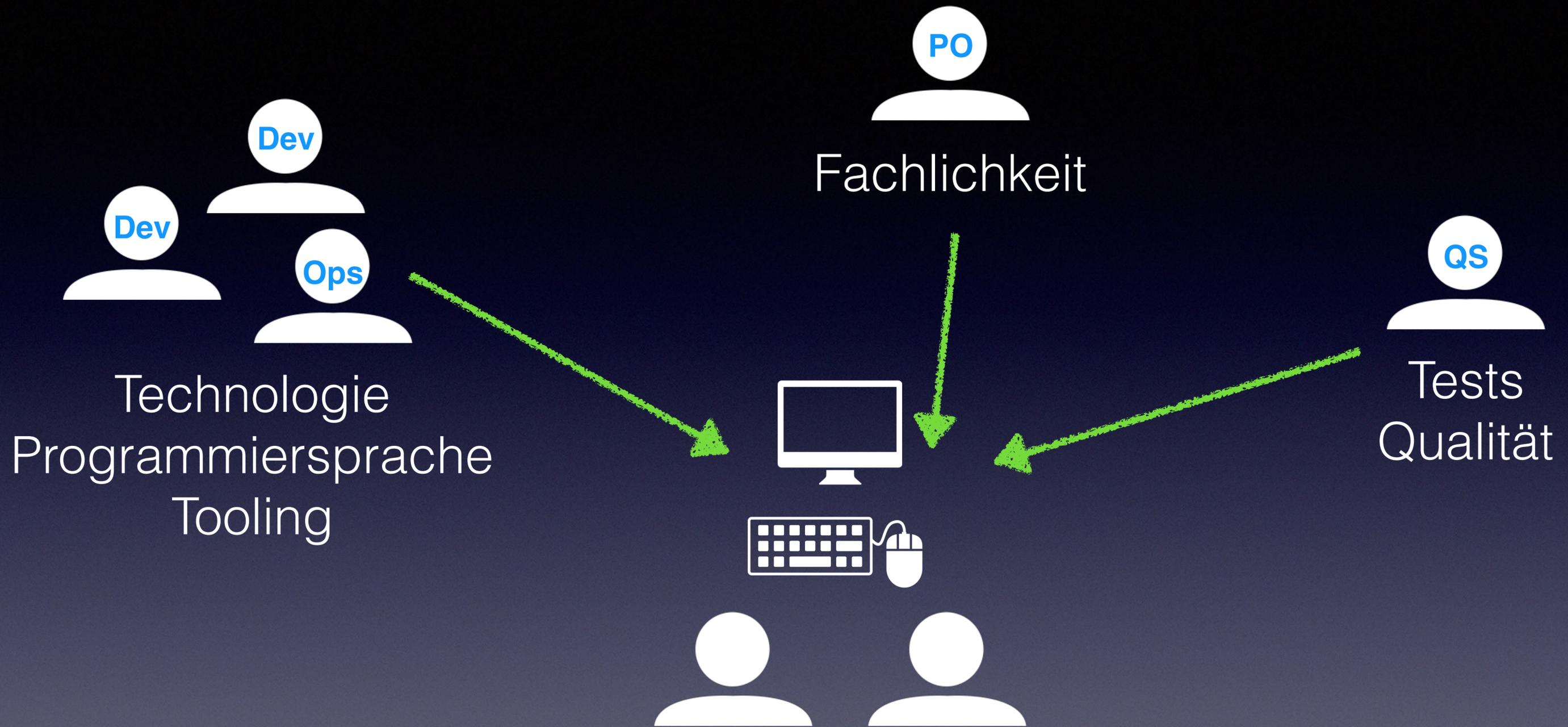
statt nachträglicher Wissensverteilung?

Technologie
Programmiersprache
Tooling

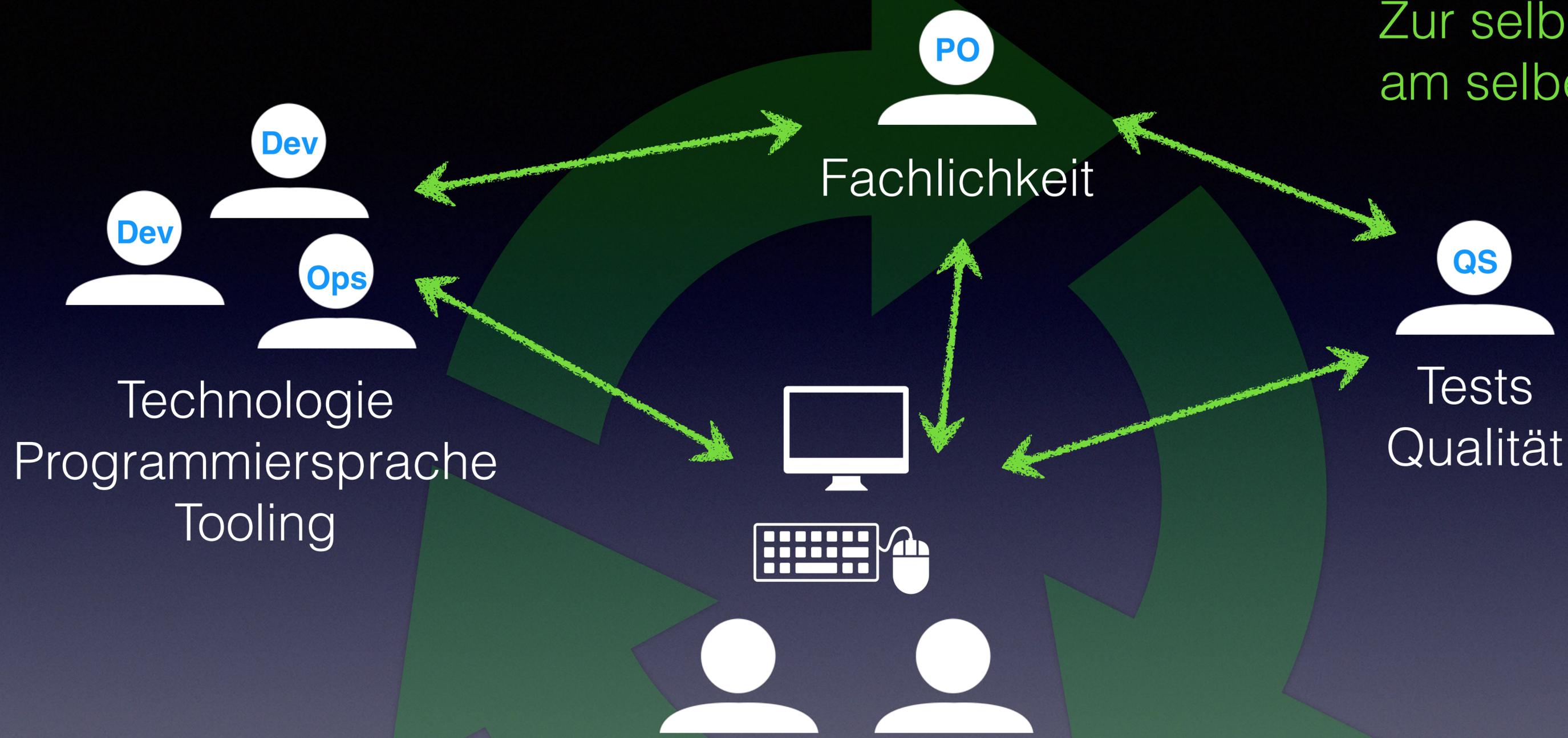
Fachlichkeit

Tests
Qualität





Zur selben Zeit
am selben Ort!



Mob Programming

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, on the same computer.“

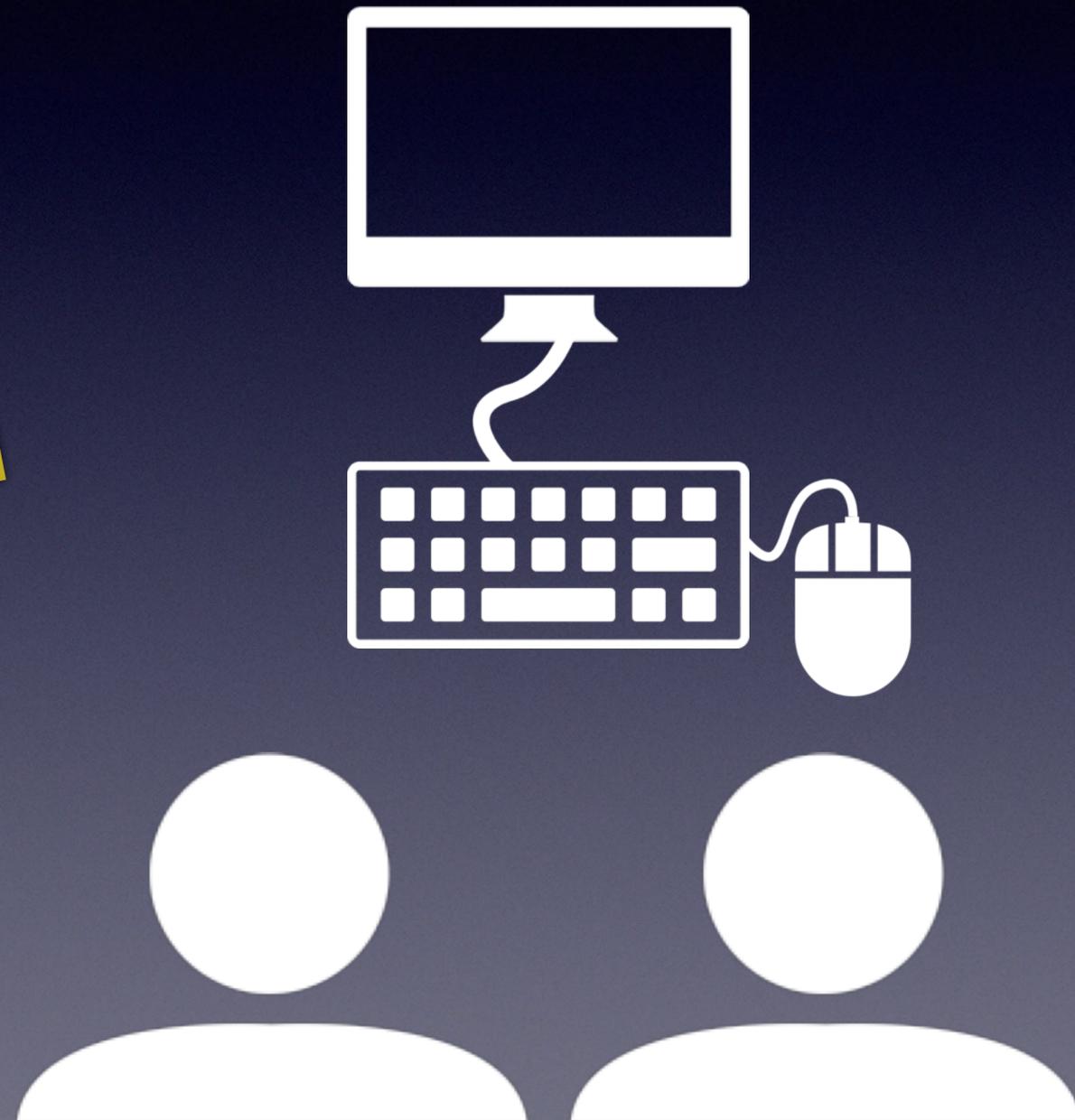
„Continuous Integration of Ideas“

– Woody Zuill

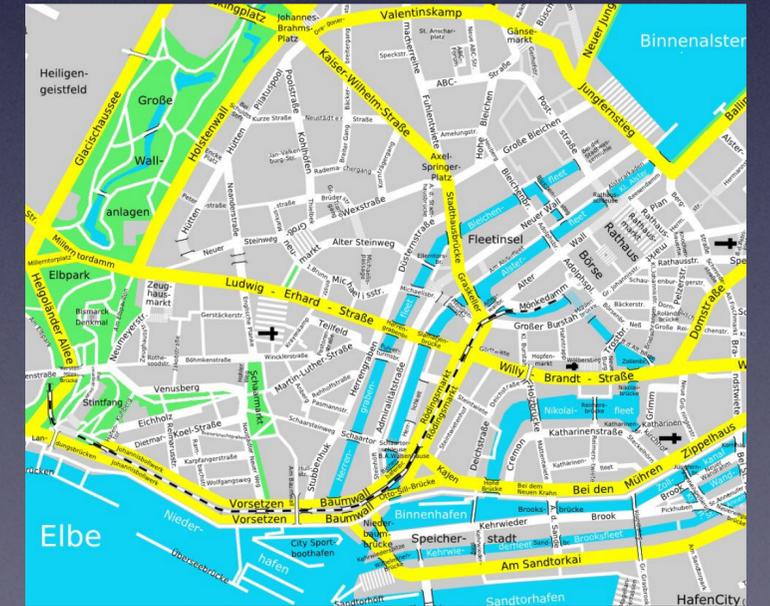
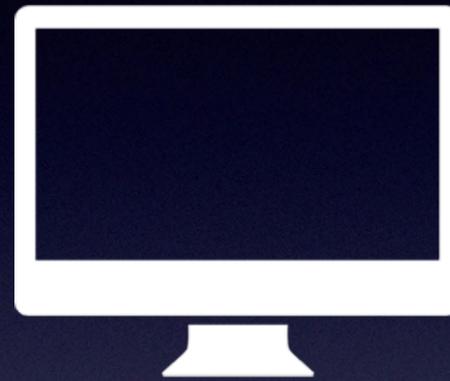
Wie funktioniert
Mob Programming?

Pair Programming (Wdh.)

1
Aufgabe



Driver & Navigator



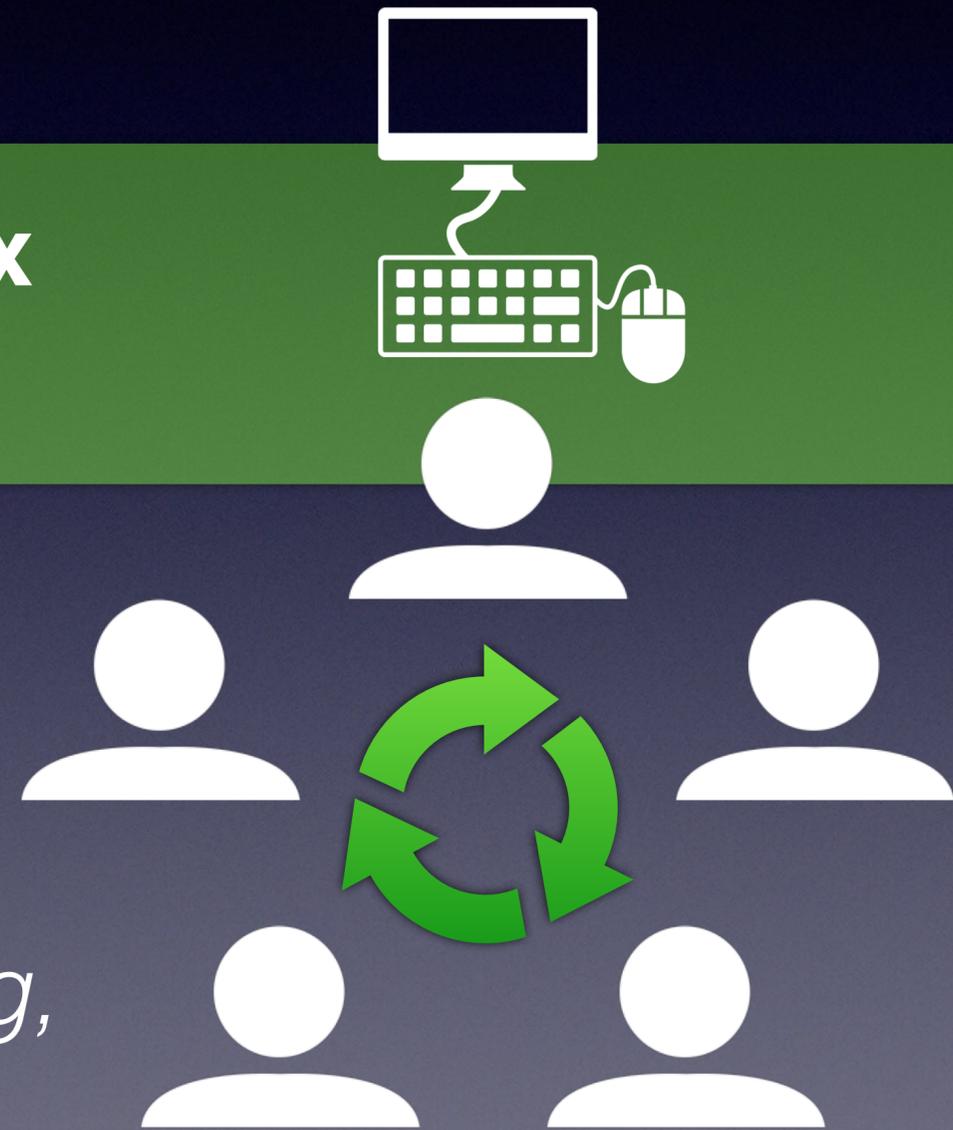
Mob Programming

Rollenwechsel!

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

*<http://mobster.cc>
oder Smartphone-Timer*

*Evtl. angenehmer als Pair Programming,
da nicht so „eng“.*

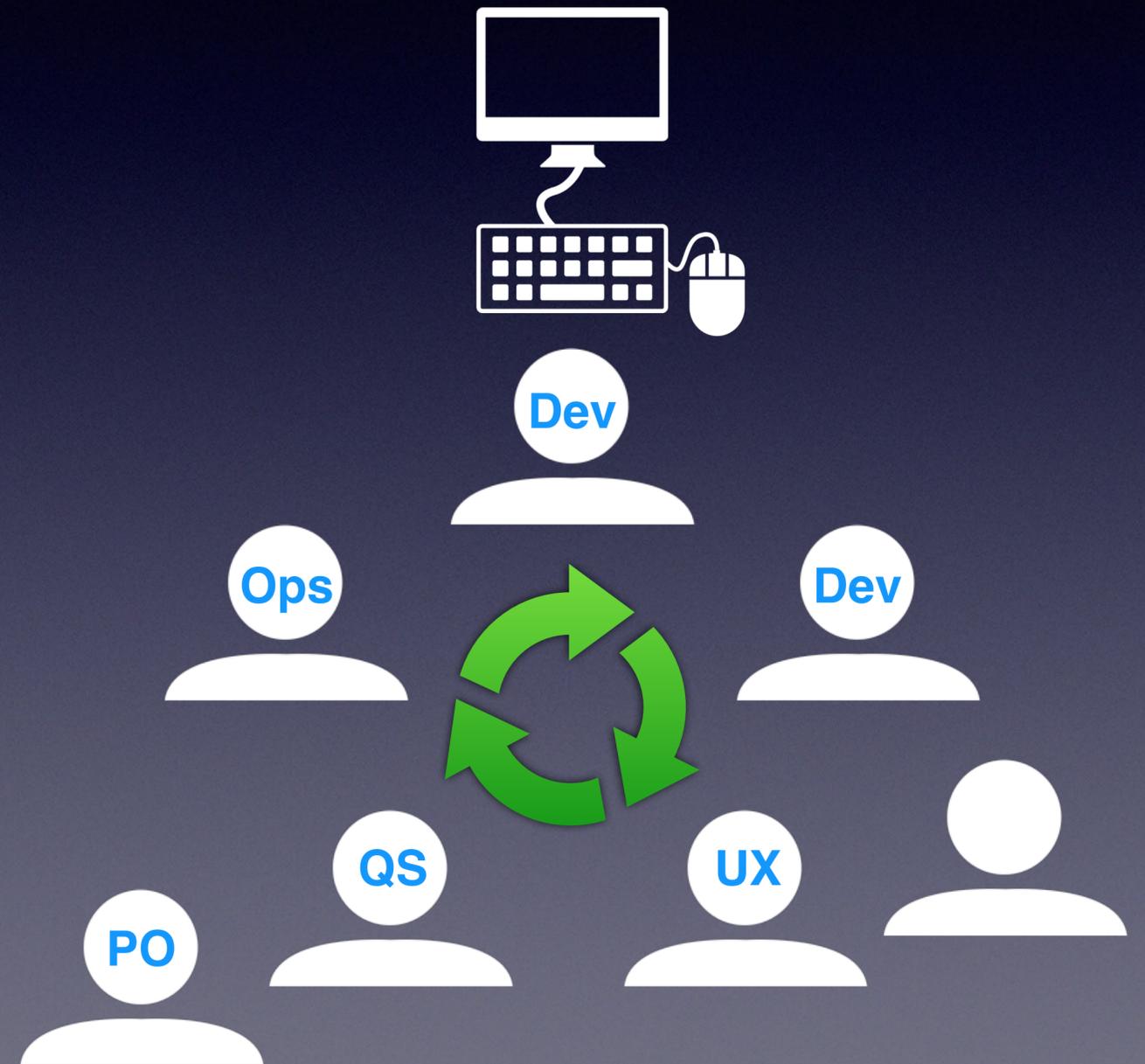


Mob Programming

*Dynamischer Mob:
Kommen und gehen.*

Rollenübergreifend!
(Aber auch Dev-only hilfreich)

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



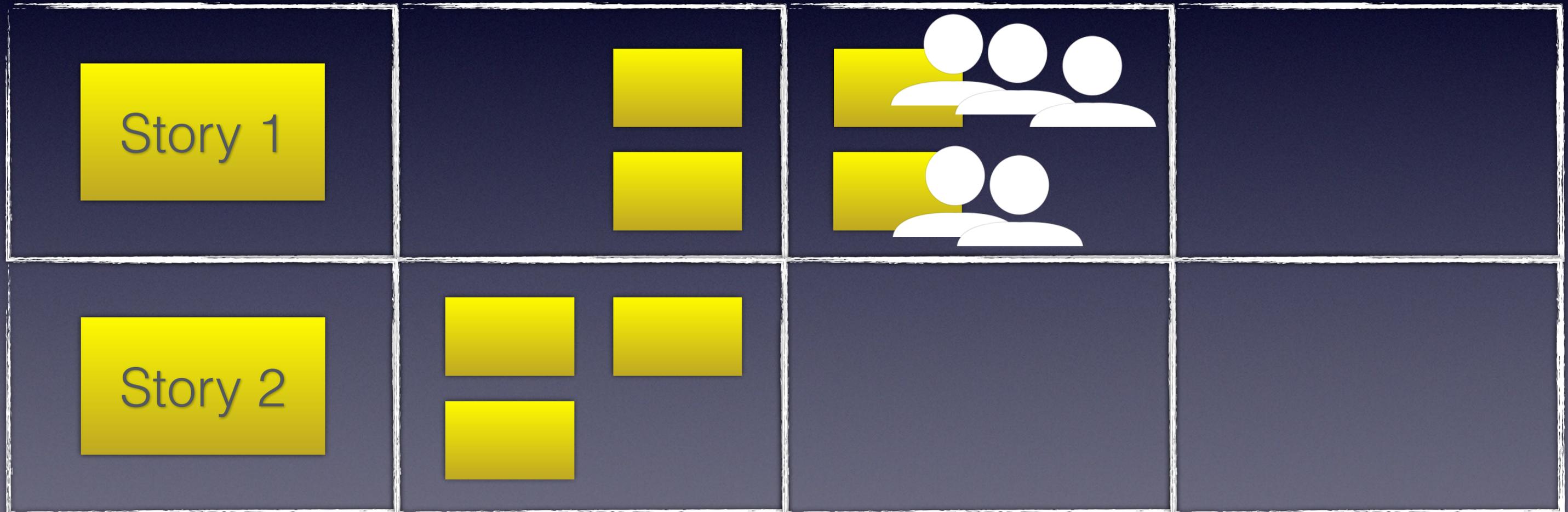
Das Wichtigste zuerst!

WIP-Limit 1

To Do

In Progress

Done



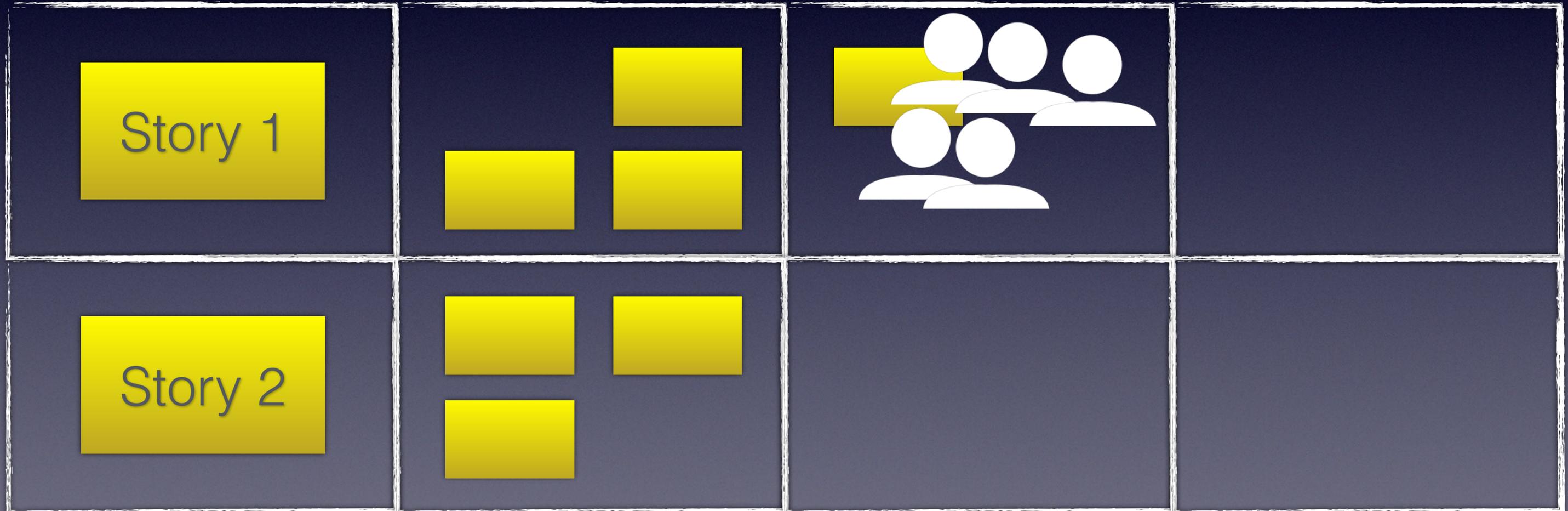
Das Wichtigste zuerst!

WIP-Limit 1

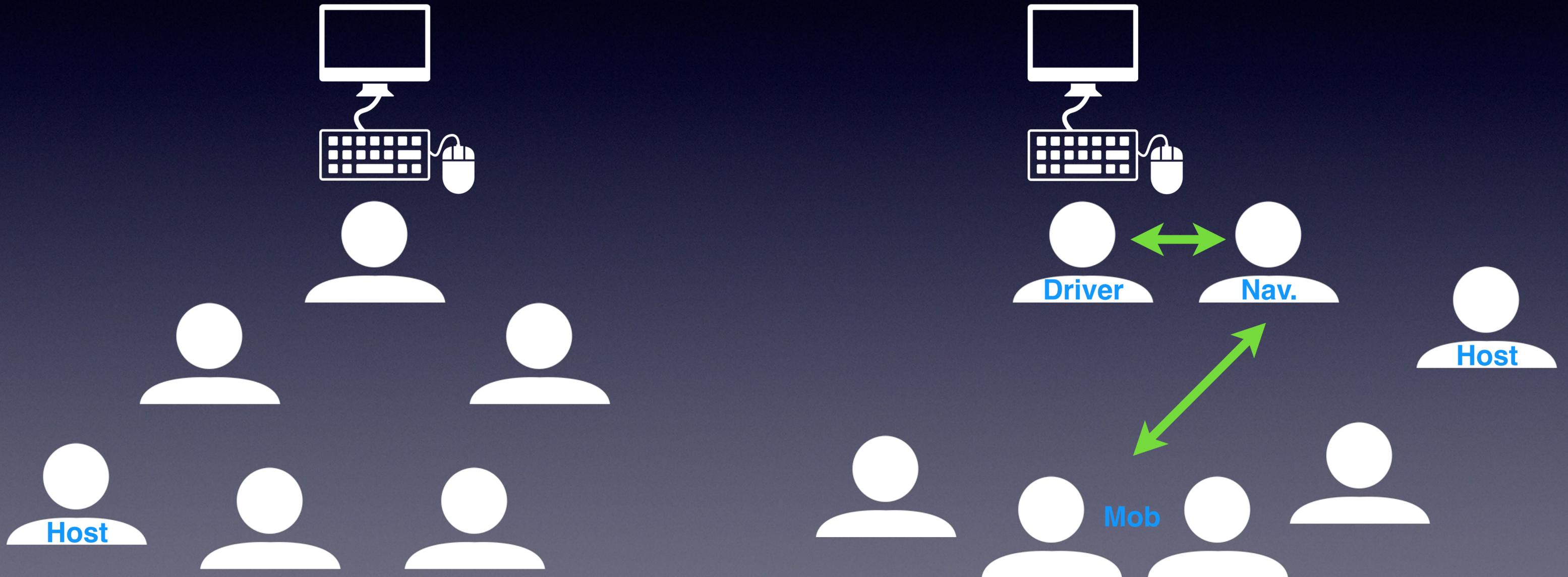
To Do

In Progress

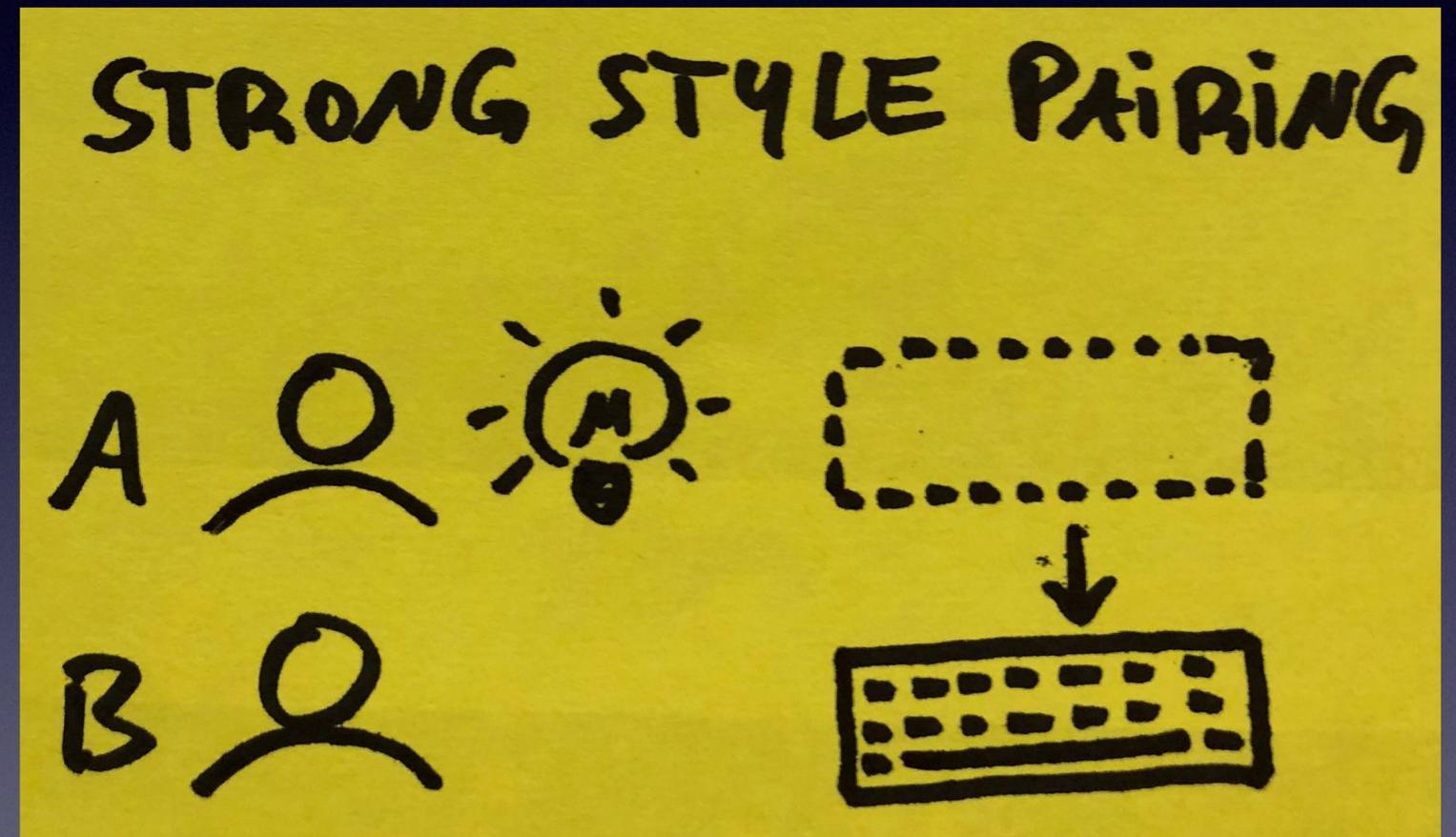
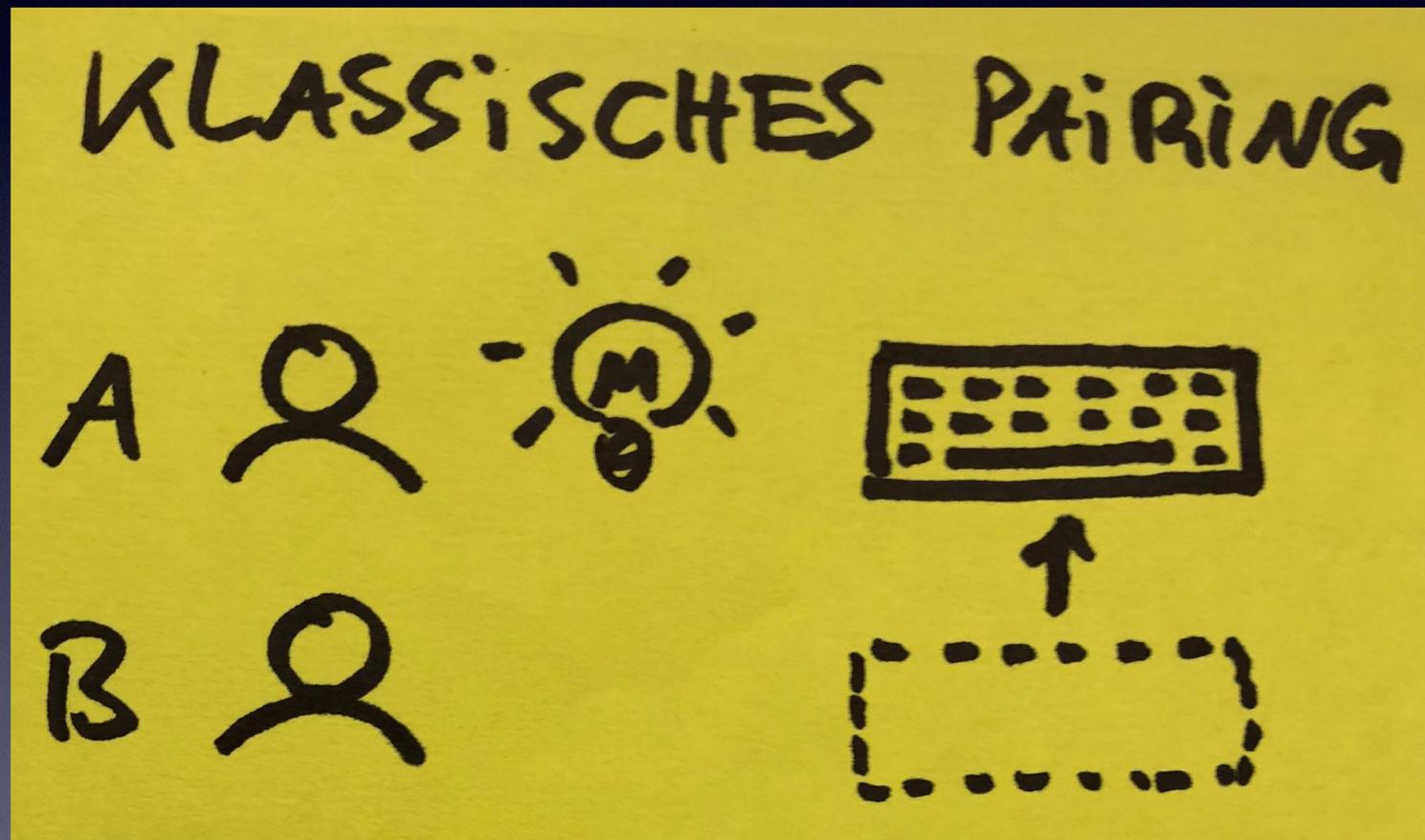
Done



Mob Programming – Setups



„Strong Style“ als Mob-Default



@LlewellynFalco

Ablauf

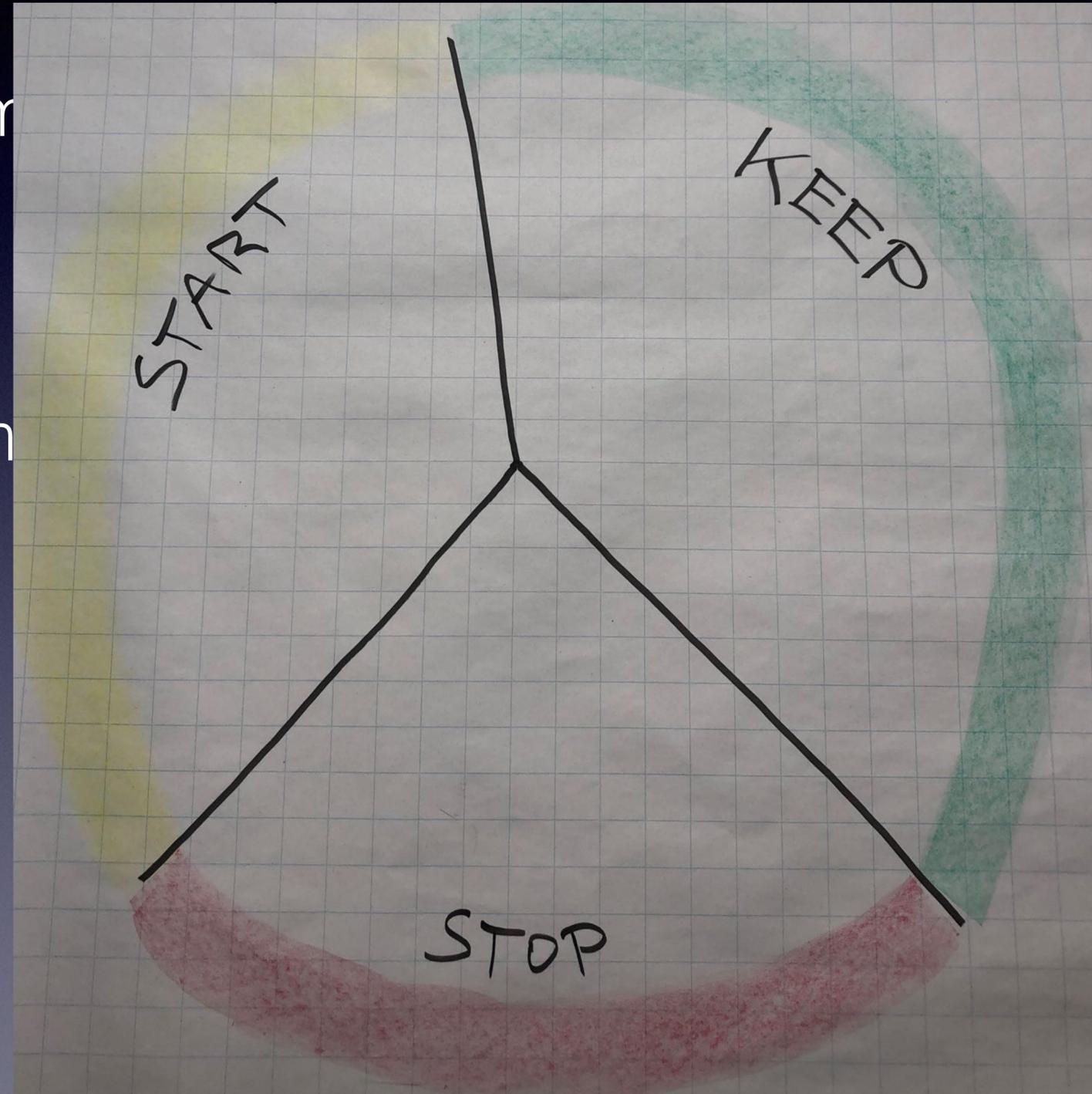
- Regeln für Mob-Session festlegen (Timebox, Navigator? etc.)
- Kurze Intro zur Problemstellung
- Discovery/Estimation/Planning/Implementation mit Tastaturwechsel
- Kurze Retro (Start/Stop/Keep)

Pause ca. 1x pro Stunde (vgl. Pomodoro beim Pairing)

Dauer je Mob-Session? 1h, ..., 1/2 Tag, 1 Tag

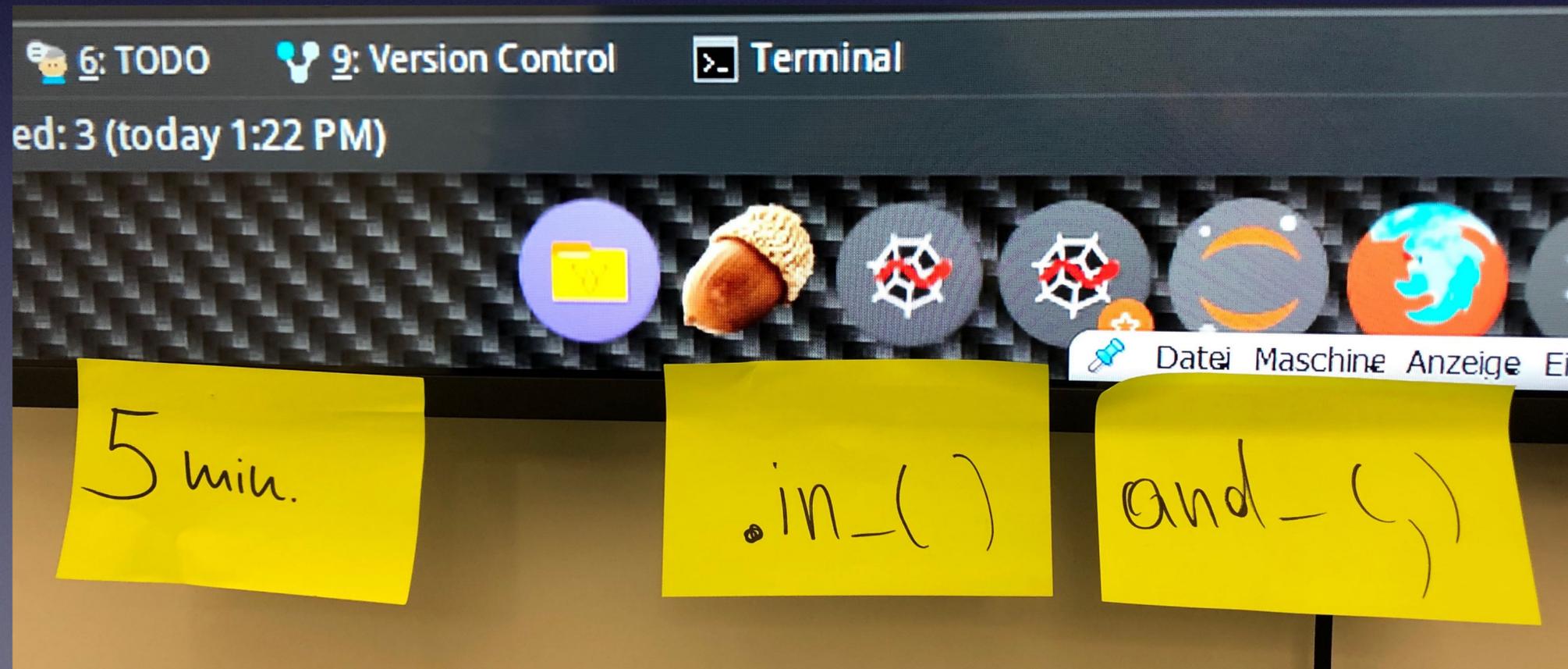
Ablauf

- Regeln für Mob-Session festlegen (Timebox)
- Kurze Intro zur Problemstellung
- Discovery/Estimation/Planning/Implement
- Kurze Retro (Start/Stop/Keep)



Hilfreich

- Regeln & Erkenntnisse sichtbar festhalten
(Timebox, Ideen, Shortcuts ...)
- *Alle Medien nutzen!*
(Flipchart, Whiteboard, Wallpaper, beschreibbare Wände, Post-Its ...)



Unsere Erfahrungen

Timebox 5 Min.

bringt Interaktion und Abwechslung

Smartphone-Timer

klares Signal, trotzdem flexibel

Mob-Größe 5-6

jeder ist häufig genug an der Tastatur

Funktastatur oder Stehpult

(mit Präferenz auf „Menschen bewegen“)

100% Mob Programming?

Nein

(wie beim Pair Programming)

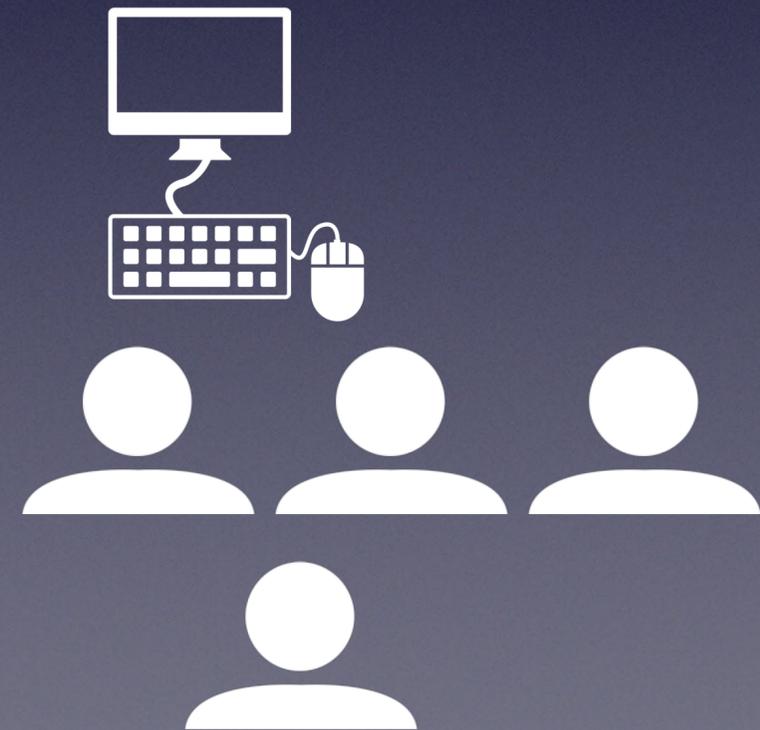
Hilfsmittel, Werkzeug

Warum üben wir das?

Werkzeuge müssen beherrscht werden,
um sie passend einsetzen zu können.

Default-Modus unter Druck (Deadline)?

Oft nicht effektives Zusammensitzen
von 3+ Devs



Warum wird das wichtig(er)?

Charakter der Probleme, die wir lösen, ändert sich.

Team muss den Lösungsweg verstehen (nicht nur das Ergebnis).

„Warum ist Code *nicht* da?“

Gemeinsam dran arbeiten. Lösungsweg gemeinsam erleben.

Cynefin

Neue, unbekannte
Technologien

Verteilte, Event-
getriebene Systeme



Komplexe Henne-Ei-Probleme

Das haben wir
noch nie gemacht.

Wie sollen wir
das schätzen?

Discovery
Estimation



Planning
Implementation

DoR

Ohne Schätzung
können wir das
nicht umsetzen...

DoD

**Mob
Programming**

Unterschied zu...

Demo vor großer Runde

Review-Meetings

Technical Grooming

Best-Practices-Meeting

PoC-Vorstellung

...

Oft nicht wirklich effizientes Zusammensitzen

Code nicht in den Händen gehabt

Code nicht wirklich verstanden

Unterschied zu Meetings

Am Ende haben wir

lauffähigen, getesteteten, **verstandenen Code**.

Mob Programming:

Zusammen lernen. Zusammen liefern.

„Mob Programming ... is the most important improvement
I've seen the last couple of years.“

– **Marcus Hammarberg**

<https://twitter.com/marcusoftnet/status/1042708243544514560>

Aber die Geschwindigkeit!

deliver ~~output~~

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

do things right
do the right things

deliver value
deliver outcome

Dürfen wir das?

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.)

[About](#)

[Meet Us](#)

[GitHub](#)

Frederik Mars / Januar 25, 2019 / [Development](#)

Mob Programming: Stateful Streaming

Am 29.09.18 fand auf dem Otto Campus ein *Never Code Alone* Event statt, bei dem sich über 30 Scala Enthusiasten trafen, um in interaktiven Mob Programming Sessions gemeinsam zu programmieren.

Insgesamt gab es vier Sessions über den ganzen Tag verteilt. Am Anfang jeder Session gab es eine kurze Einleitung zum Thema. Anschließend wurde dem ersten Mutigen im Publikum eine Funktastatur überreicht und es ging los mit dem Mob Programming.

Meetup

29
MAI

Vergangene Events

Mob Programming bei Kühne+Nagel



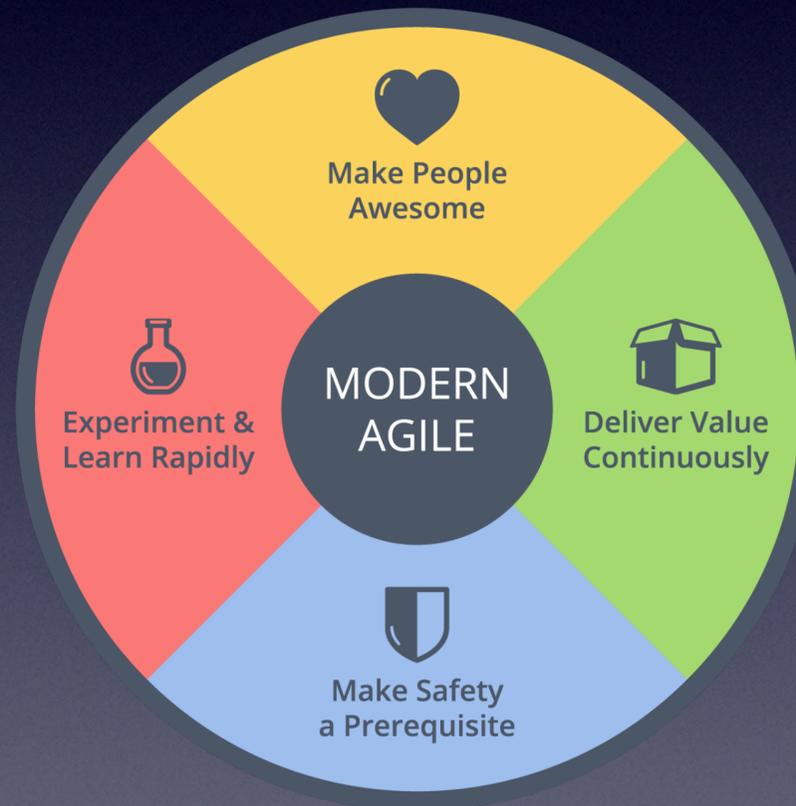
Veranstaltet von Sven Bunge und 3 anderen

Von [Softwerkskammer Hamburg](#)

Öffentliche Gruppe ?

Modern Agile

Agile – aber so, dass es funktioniert.



#MobProgramming

#NoEstimates

#NoProjects

#ModernAgile

<http://modernagile.org/>

#RemoteMobProgramming

<https://www.remotemobprogramming.org>



by @simonharrer, @jochen_christ & @Waterback

Mob Programming

Pair Programming

Know-How-Transfer

Coaching

Pomodoro

XP

Fragen?

Lesbarkeit

Modern Agile

Einfachheit

Strong Style Pairing

Geschwindigkeit

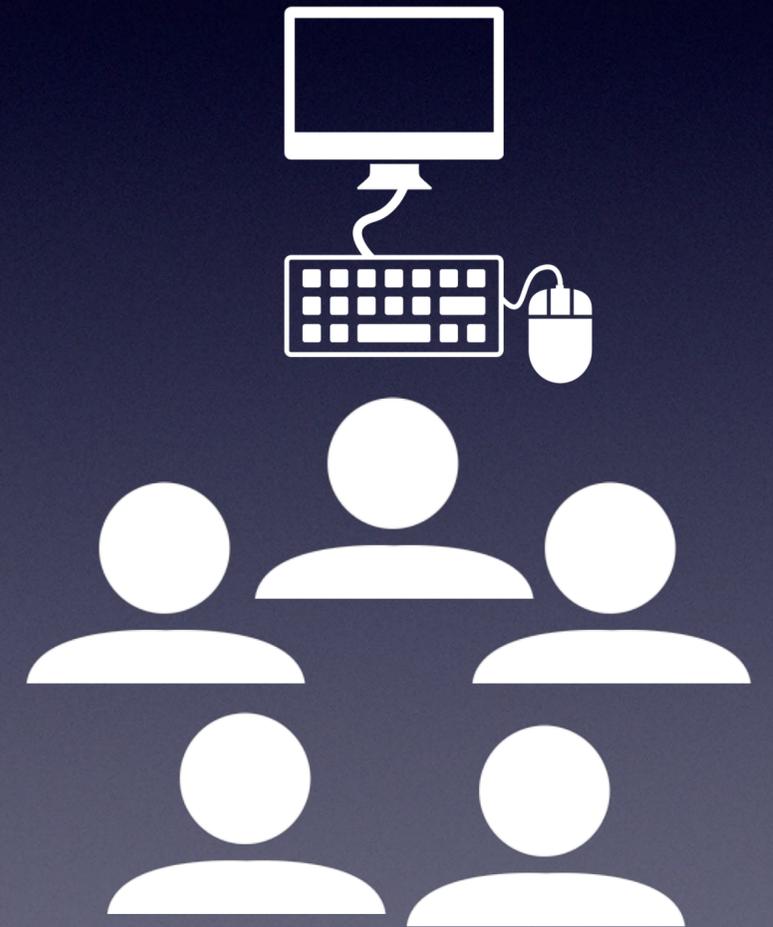
TDD

Collective Product Ownership

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

Danke!

Thomas Much
 @thmuch



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