

# Fast fertig, Big Bang, Hausdurchsuchung und 7 weitere agile Fails

**SCRUMDAY 2019**

“Scrum ist ein Hilfsmittel  
die Welt zu verändern”



Timon Fiddike & Sebastian Lange

@TimonFiddike & @agilebastian

# **Dot Voting JETZT an der Pinnwand!**

## **Platziere 3 Punkte: Wir haben Zeit für die top 4–5 Fails**

- 1. BIG-BANG: WECHSEL ZU FEATURE TEAMS**
- 2. ABBRUCH NACH 1 M€, NICHTS DAZU GELERNT**
- 3. EIN DRAMA IN 4 AKTEN, ABER MIT HAPPY END: LOVOO & LeSS**
- 4. ESKALATION AN DEN CEO: “COACHES” SCHLAGEN EINE LÖSUNG VOR**
- 5. COMMAND, CONTROL & WETTRENNEN: ZIELSETZUNG MIT OKR**
- 6. SPÄT GERETTET DURCH “NEUE” IDEE**
- 7. CLOUD MIGRATION: ES IST NICHT EINFACH NUR EIN UMZUG**
- 8. ALLE ENTWICKLER VERLOREN: PROZESSHANDBUCH und “STAR” TEAM**
- 9. DANN LASS UNS EINE TASK FORCE GRÜNDEN!**

Timon Fiddike & Sebastian Lange



@TimonFiddike & @agilebastian

WECHSEL ZU FEATURE TEAMS

**BIG-BANG**

“

In this case, one usually sees a planning spreadsheet similar to the following:

”

| Feature   | Component<br>A | B | C | D | E | ... |
|-----------|----------------|---|---|---|---|-----|
| Feature 1 | x              | x |   |   | x |     |
| Feature 2 | x              | x | x |   |   |     |
| ...       |                |   |   |   |   |     |

Quelle: Craig Larman and Bas Vodde, 2008. Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum (1 ed.). Addison-Wesley Professional.



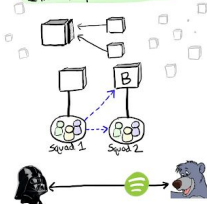
# Spotify Engineering Culture

Part 1 of 2  
Henrik Kniberg  
Jan 2016

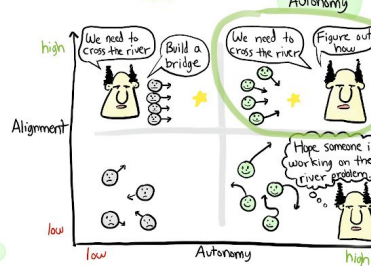
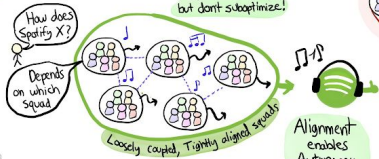
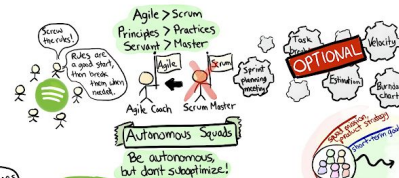
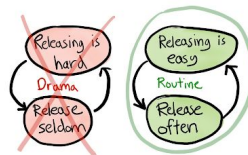
Cross-pollination > Standardization



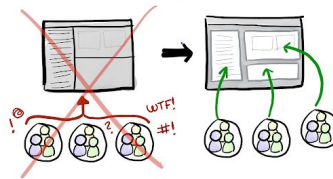
Internal Open-source model



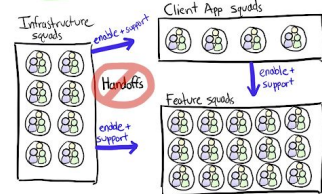
Small + frequent releases



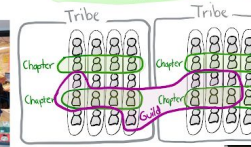
Decoupled releases



Self-service model enable > serve

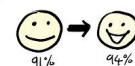


Community > Structure



Focus on Motivation

Hi everyone,  
Our employee satisfaction survey says  
**91% enjoy working here,**  
and 4% don't.  
**This is of course not satisfactory,**  
and we **want to fix it.**  
If you're one of those unhappy 4%,  
please contact us.  
We're here for your sake, and nothing else.



Trust > Control

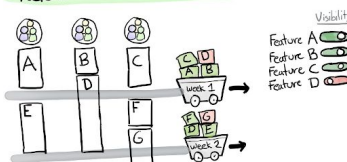


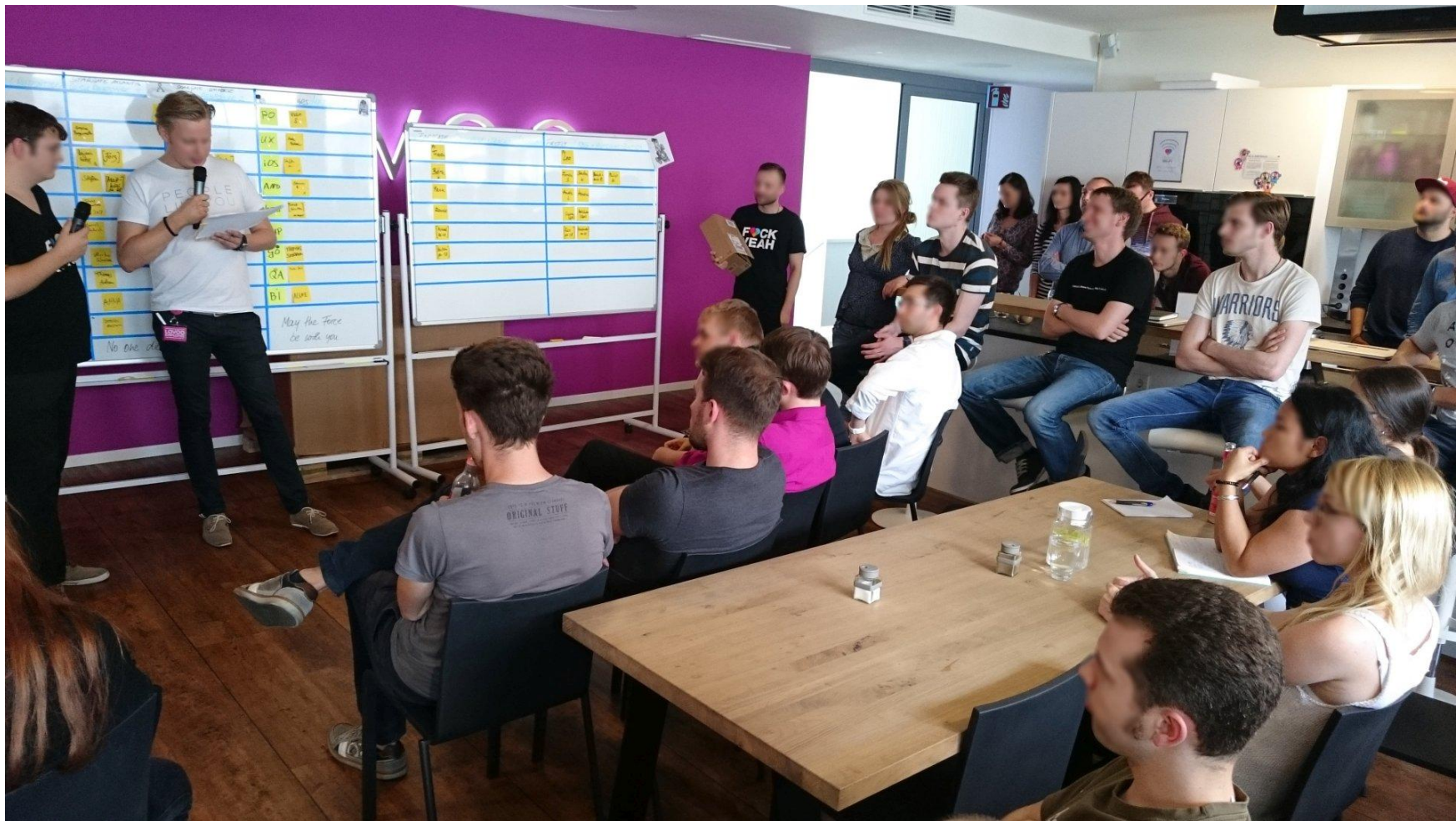
People > \*

My colleagues are awesome!



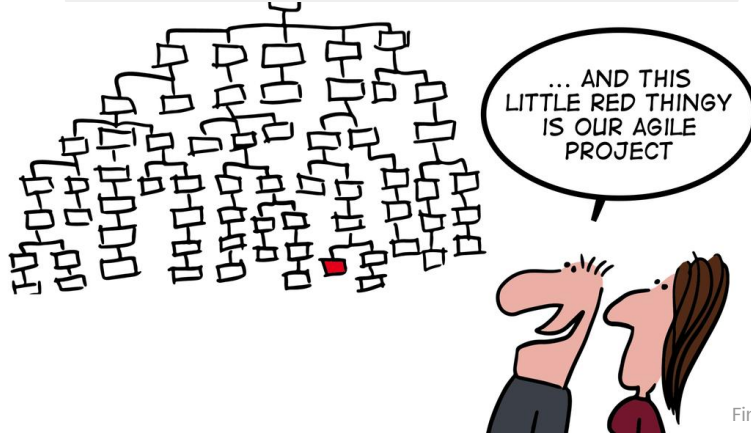
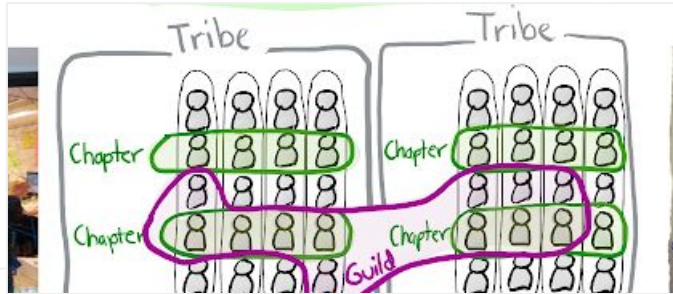
Release Trains + Feature Toggles







# Learnings



- Vorsicht bei Copy & Paste und bei Begriffen  
"Spotify Model", "Tribe", "Squad", "Chapter", ...
- Die Wurzel der cross-funktionalen  
Zusammenarbeit adressieren
- Globale Optimierung statt Lokaler Optimierung  
(nur Tech ändert sich)

ZIELSETZUNG MIT OKR

# COMMAND, CONTROL & WETTRENNEN

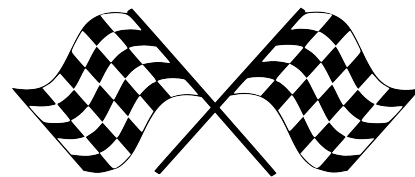
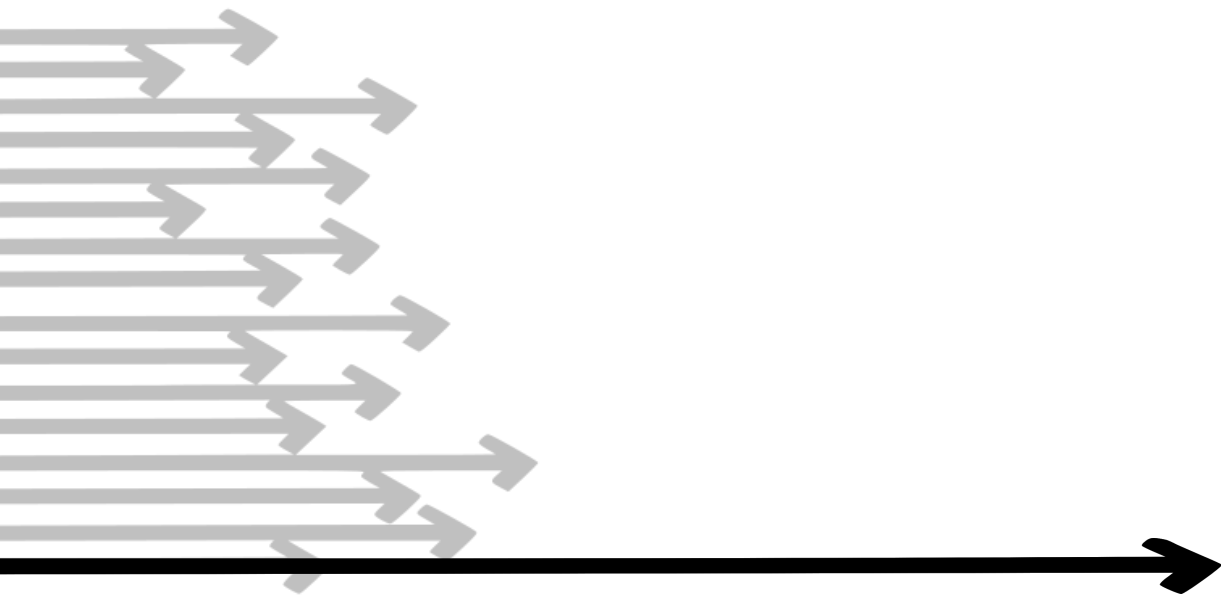
# Elements of an OKR

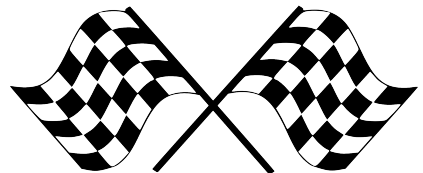
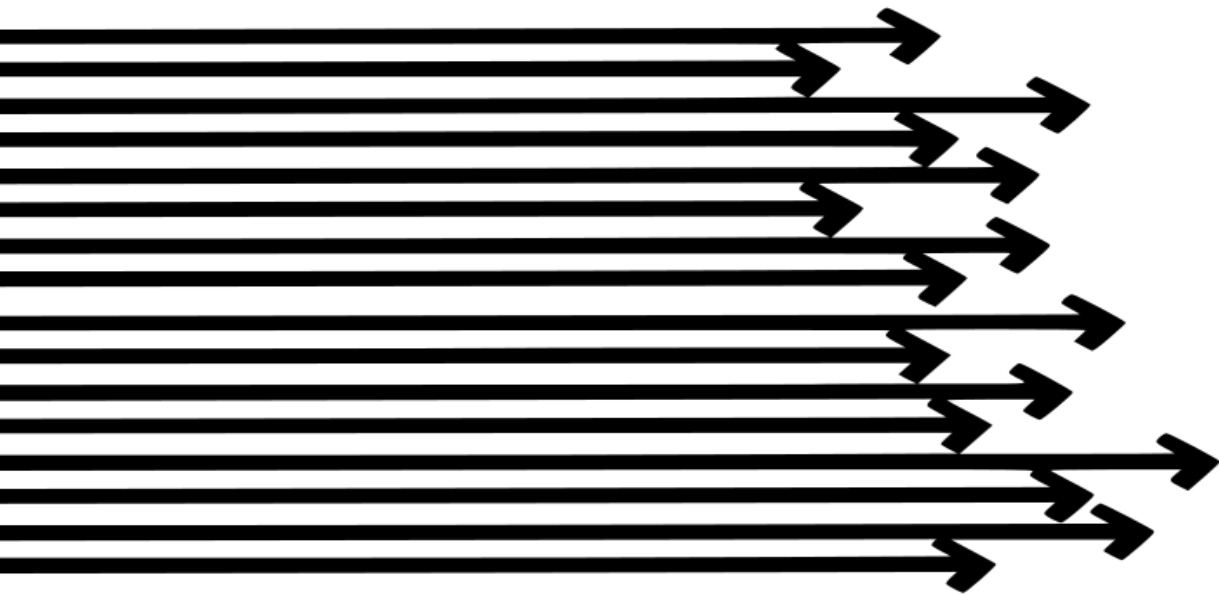
- The Objective...
  - is ambitious
  - feels a tad uncomfortable
- The Key Results
  - clearly make the objective achievable
  - are quantifiable
  - lead to objective grading

Google ventures<sup>+</sup>  
**STARTUP LAB**

- “Objectives and Key Results”
- Ursprung von Intel aus den '70ern
- bekannt vor allem von Google
- Management- / Führungsmethode
- Zielsetzung in mittelfristigen Timeboxes (z.B. 90 Tage, ½ Jahr, 1 Jahr)

Quelle: <https://www.businessinsider.com/googles-ranking-system-okr-2014-1?IR=T>





????????



**Workpath**  
**Alignment**  
**SMART**  
**Transparency**  
**Update**  
**Progress**  
**Goalsetting**  
**Key-Results**  
**Objectives**  
**Milestones**  
**Bergfest**



# Learnings

**Ahaaa!**



- Expertenwissen & aktive Betreuung hilft :)
- Start with the Why : welches Problem wollen wir denn lösen?  
Und ist Framework XYZ überhaupt das Richtige Werkzeug?
- Features sind schlechte Objectives
- Kommunikation ist der Schlüssel:  
“macht gute OKR” vs. “wir setzen uns alle Ziele”

NEUES BACKEND

**ABBRUCH NACH 1 M€, NICHTS  
DAZU GELERNT**

# Kontext



- Startup
  - < 5 Jahre
  - ca. 200 Mitarbeiter
  - XX € Investment eingeworben

# Start



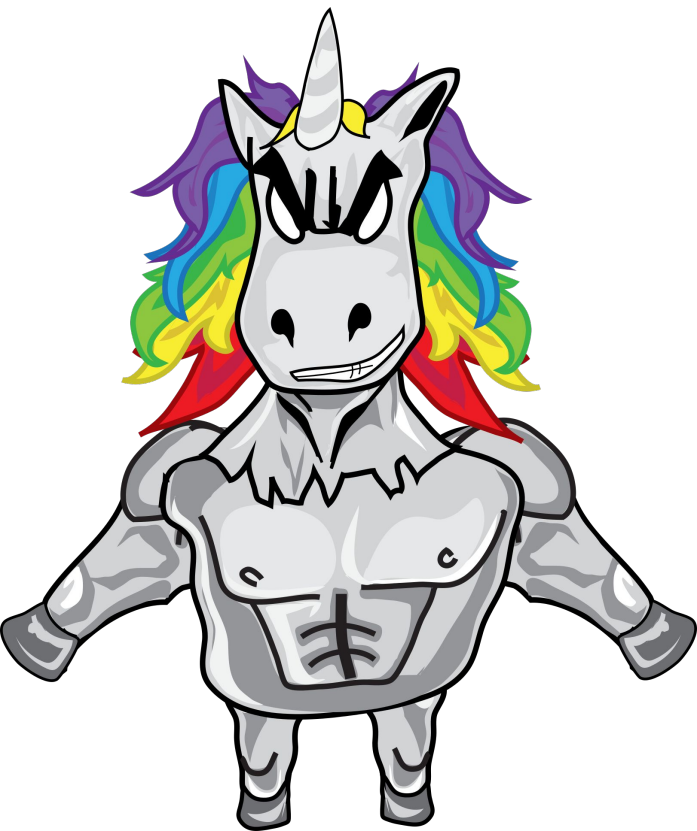
- CEO: “Große wichtige Feature Idee!”

# Start



- CEO: “Große wichtige Feature Idee!”
- Team:
  - “Das alte Backend ist total fies: Rewrite!”
  - “Mit dem alten Backend geht das Feature nicht!”

# Start



- CEO: “Große wichtige Feature Idee!”
- Team:
  - “Das alte Backend ist total fies: Rewrite!”
  - “Mit dem alten Backend geht das Feature nicht!”
- CTO: “Wir bauen neues Backend incl. neues Feature!”

# Start

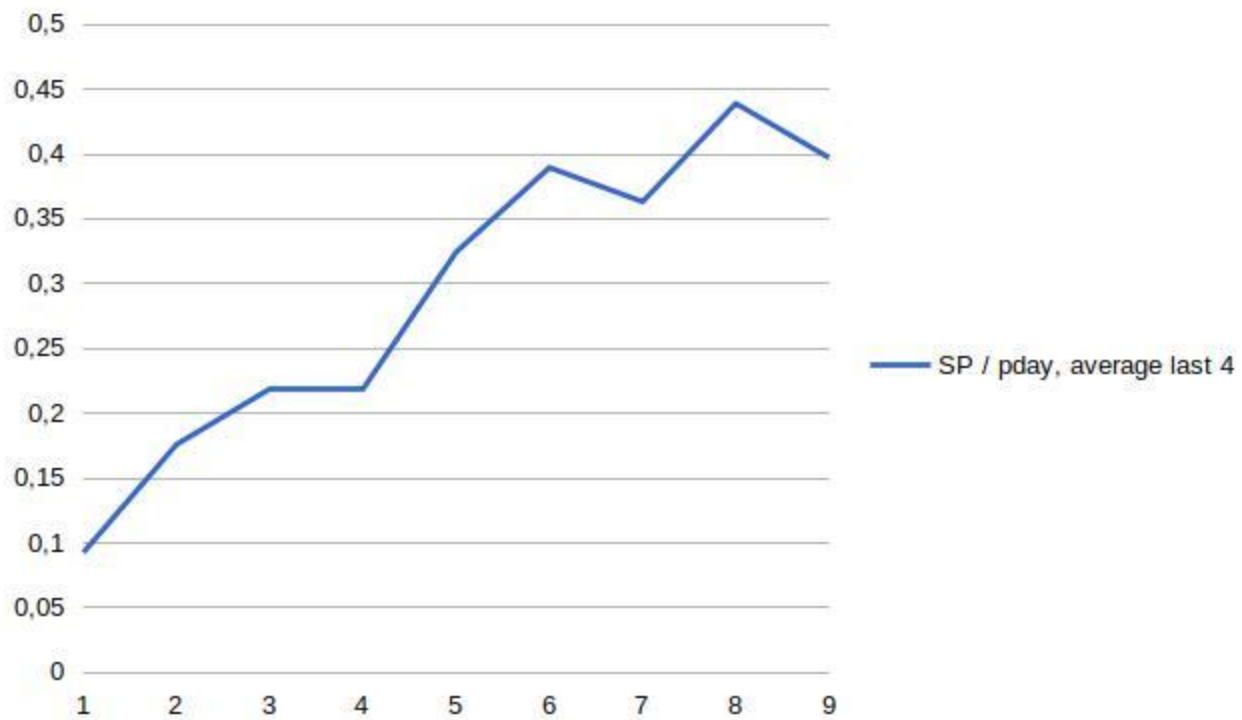


- CEO: “Große wichtige Feature Idee!”
- Team:
  - “Das alte Backend ist total fies: Rewrite!”
  - “Mit dem alten Backend geht das Feature nicht!”
- CTO: “Wir bauen neues Backend incl. neues Feature!”
- “Coach”:
  - “Rewrite geht in 92 % der Fälle schief, lieber
    - neues Feature ins alte Backend
    - das alte iterativ verbessern wo nötig.”

# Start



- CEO: “Große wichtige Feature Idee!”
- Team:
  - “Das alte Backend ist total fies: Rewrite!”
  - “Mit dem alten Backend geht das Feature nicht!”
- CTO: “Wir bauen neues Backend incl. neues Feature!”
- “Coach”:
  - “Rewrite geht in 92 % der Fälle schief, lieber
    - neues Feature ins alte Backend
    - das alte iterativ verbessern wo nötig.”
  - “Naaa gut, dann helfe ich Euch halt das alles mit Scrum zu schaffen.”
  - “Empfehlung: CTO als PO.“



Bitte Hand heben: Wer glaubt ...

1. ... nach 1 Jahr war das neue Backend live?

Bitte Hand heben: Wer glaubt ...

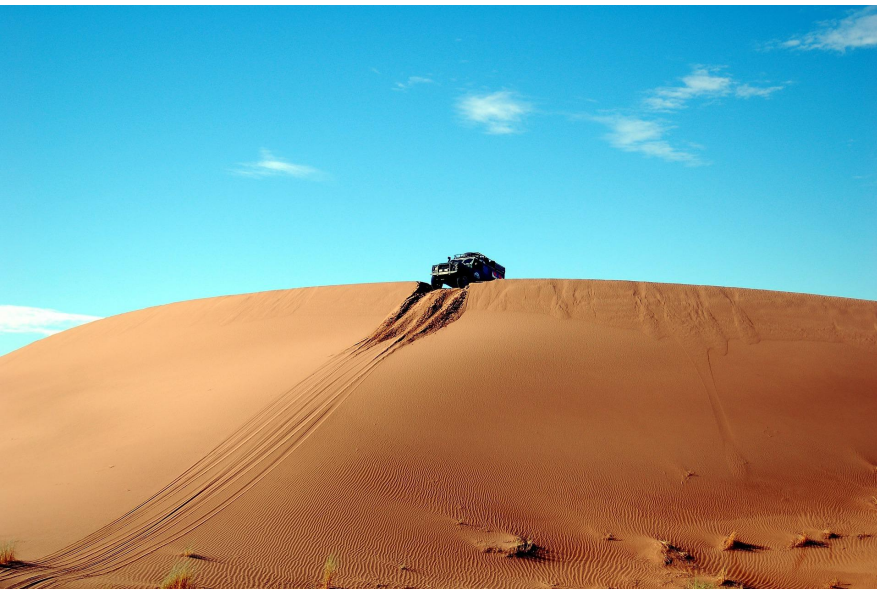
1. ... nach 1 Jahr war das neue Backend live?
2. ... nach 1 Jahr war das neue Feature live?

# 1 Jahr später



- CEO: "Macht das überhaupt noch Sinn???"
- PO=CTO: "O.K., wir prüfen mal die geschäftliche Hypothese."

# 1 Jahr später



- CEO: "Macht das überhaupt noch Sinn???"
- PO=CTO: "O.K., wir prüfen mal die geschäftliche Hypothese."
- Team:
  - "O.K., wir bauen wertvollsten Spezialfall, hardcoded."

# 1 Jahr später



- CEO: “Macht das überhaupt noch Sinn???”
- PO=CTO: “O.K., wir prüfen mal die geschäftliche Hypothese.”
- Team:
  - “O.K., wir bauen wertvollsten Spezialfall, hardcoded.”
  - “Oh. Kunden verwenden das Feature nicht.”

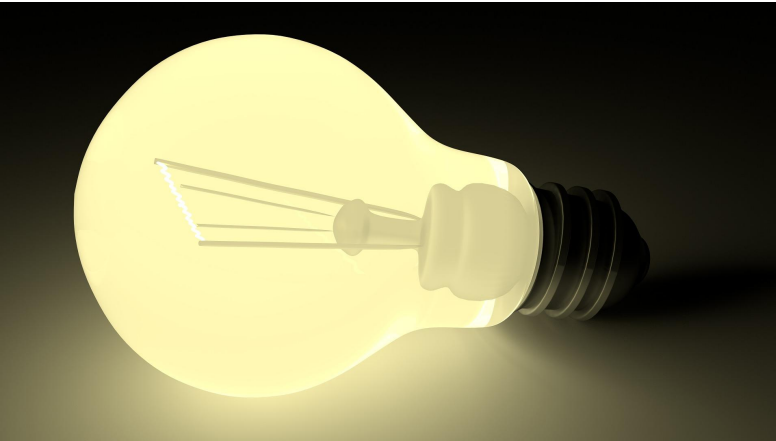
# 1 Jahr später



- CEO: “Macht das überhaupt noch Sinn???”
- PO=CTO: “O.K., wir prüfen mal die geschäftliche Hypothese.”
- Team:
  - “O.K., wir bauen wertvollsten Spezialfall, hardcoded.”
  - “Oh. Kunden verwenden das Feature nicht.”
- (Initiative wird gestoppt, CTO verlässt Firma)

ABBRUCH NACH 1 M€ UND NICHTS DAZU GELERNT: NEUES BACKEND

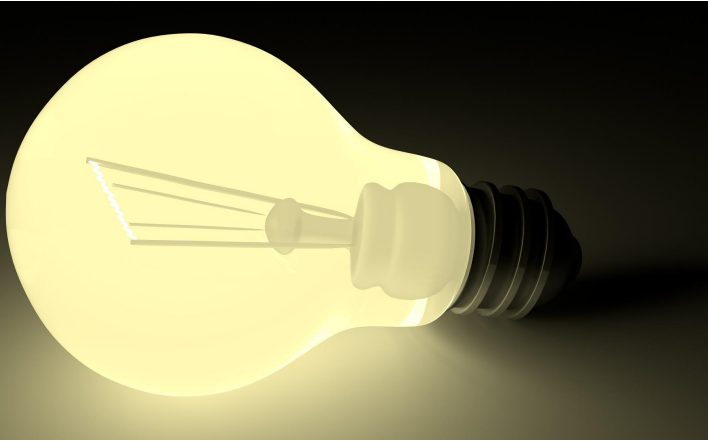
# Learnings



- Geschäftliche Hypothese prüfen!
- Scrum zum Lernen von *Was* und *Wie!*

ABBRUCH NACH 1 M€ UND NICHTS DAZU GELERNT: NEUES BACKEND

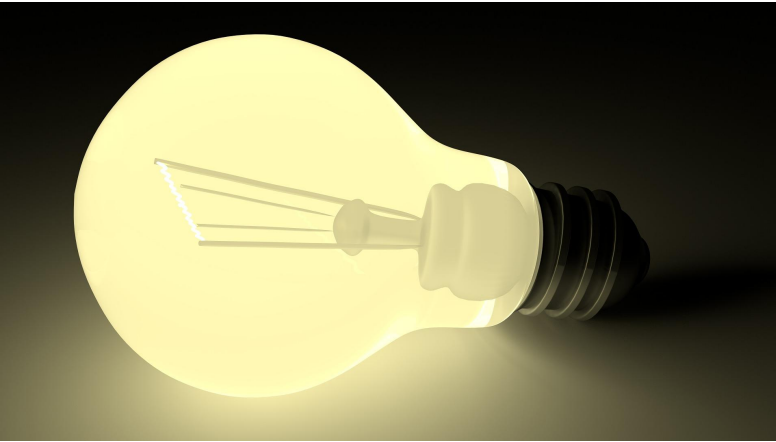
# Learnings



- Geschäftliche Hypothese prüfen!
- Scrum zum Lernen von *Was* und *Wie*!
- ~~Rewrite~~ -> Emerging Architecture, TDD, Refactoring

ABBRUCH NACH 1 M€ UND NICHTS DAZU GELERNT: NEUES BACKEND

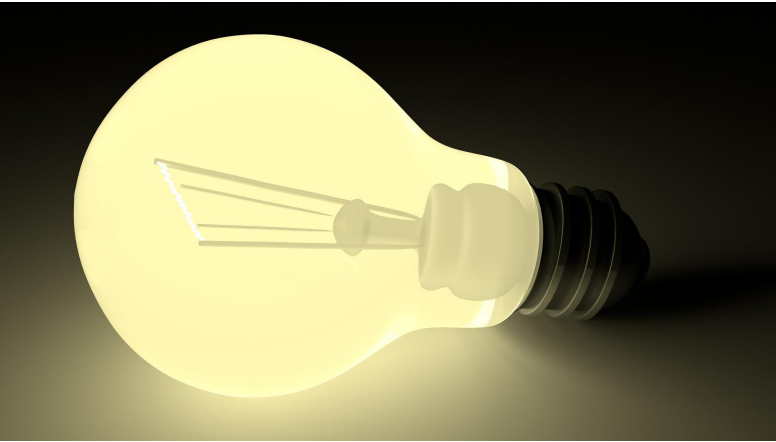
# Learnings



- Geschäftliche Hypothese prüfen!
- Scrum zum Lernen von *Was* und *Wie*!
- ~~Rewrite~~ -> Emerging Architecture, TDD, Refactoring
- CTO
  - unterwegs sehr gut als PO
  - Große Lernchance verpasst, als er Firma verlässt

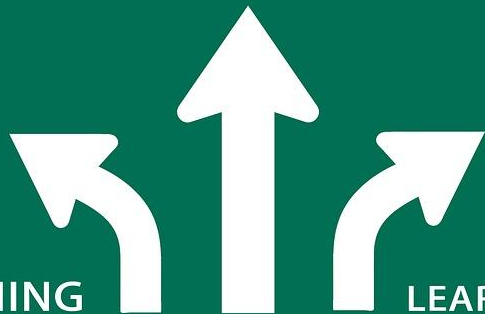
ABBRUCH NACH 1 M€ UND NICHTS DAZU GELERNT: NEUES BACKEND

# Learnings



- Geschäftliche Hypothese prüfen!
- Scrum zum Lernen von *Was* und *Wie*!
- ~~Rewrite~~ -> Emerging Architecture, TDD, Refactoring
- CTO
  - unterwegs sehr gut als PO
  - Große Lernchance verpasst, als er Firma verlässt
- Mein wichtigstes Ziel als Coach: Lernen ermöglichen!!!

LEARNING



LEARNING

LEARNING

LEARNING



# Wachstum als Coach



**Lernen ermöglichen**, dazu z.B. fragen

- An welchen konkreten Beobachtungen erkennt Ihr / Eure Kunden, wenn die Initiative Erfolg hat?
- Wann könnt Ihr/Eure Kunden das beobachten?

# Wachstum als Coach



**Lernen ermöglichen**, dazu z.B. fragen

- An welchen konkreten Beobachtungen erkennt Ihr / Eure Kunden, wenn die Initiative Erfolg hat?
- Wann könnt Ihr/Eure Kunden das beobachten?
- Wer wird wahrscheinlich vorher ungeduldig?
  - Wie könnten wir früher am Markt testen?

# Wachstum als Coach



**Lernen ermöglichen**, dazu z.B. fragen

- An welchen konkreten Beobachtungen erkennt Ihr / Eure Kunden, wenn die Initiative Erfolg hat?
- Wann könnt Ihr/Eure Kunden das beobachten?
- Wer wird wahrscheinlich vorher ungeduldig?
  - Wie könnten wir früher am Markt testen?
- Wer will aus dem nächsten Schritt lernen?
- Wer will aus der gesamten Initiative lernen?
- Wie stellen wir sicher, dass das wirklich passiert?

“COACHES” SCHLAGEN EINE LÖSUNG VOR

# ESKALATION AN CEO

# Kontext

- Startup
  - < 5 Jahre
  - ca. 100 Entwickler

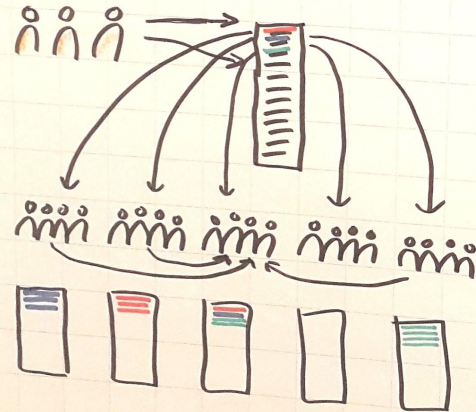
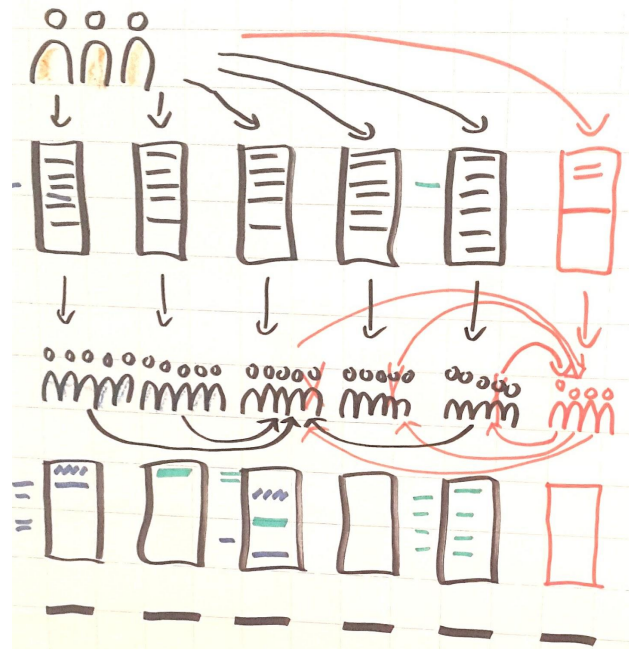
Bitte Hand heben:

Wer von Euch hat schon einmal  
bei Kunden / in der eigenen Firma

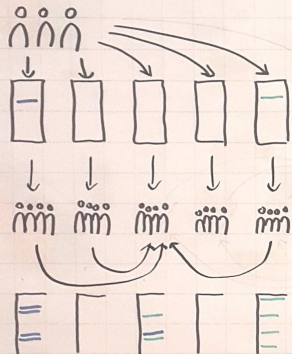
Dinge gesehen, die schon beim Zuschauen weh tun?







# Flow of Priority



Separate  
POD  
OKRs

Open  
Company  
priorities  
adapt to  
POD skills

Code Yolo  
+ projects  
are blocked  
for urgent  
goals

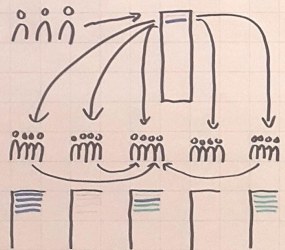
Projects  
are slowed  
down by  
dependencies

Projects  
+ Code Y.  
slow down  
PODs.

unclear  
(cross)  
priorities

risk of  
delay  
of urgent  
items

frequent  
escalation  
to top  
management



"All POD  
functioning  
under  
one list"

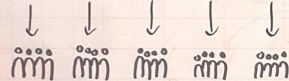
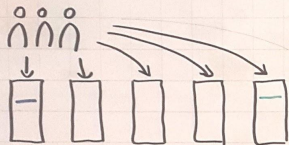
POD skills  
adapt to  
company  
priorities  
with minimal

Each POD  
priorities  
are  
clear

1st Step:  
Implement  
a  
shared group  
of PODs

AGILE.COACH

# Flow of Priority



Separate  
POD  
OKRs

Open to  
Company  
priorities  
adapt to  
POD skills

Code Yell  
+ projects  
are checked  
for urgent  
goals

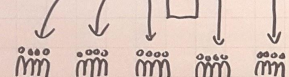
Projects  
are slowed  
down by  
dependencies

Projects  
+ Code Y.  
slow down  
PODs.

unclear  
(cross)  
priorities

risk of  
delay  
of urgent  
items

Frequent  
escalation  
to top  
management



"All POD  
Backlog/  
ordered list"

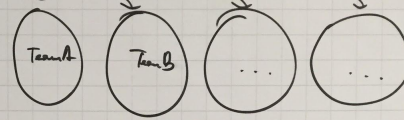
POD Skills  
adapt to  
Company  
priorities  
via matrix

Cross POD  
priorities  
are  
clear

1st Step:  
Implement  
for  
select group  
of PODs

AGILE.COACH

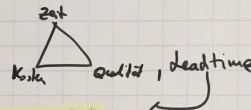
① all POD Backlog/  
ordered list



② WIP:

a) Quality, Throughput

b) Cost of Delay



③ Pull vs Push

Pull → allows quality standards for complex product

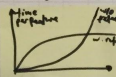
Push → leads to shortcuts (costly in the mid-term)

④ Technische Schuld / Qualität

→ Big Bang quality improvement / tech debt invest

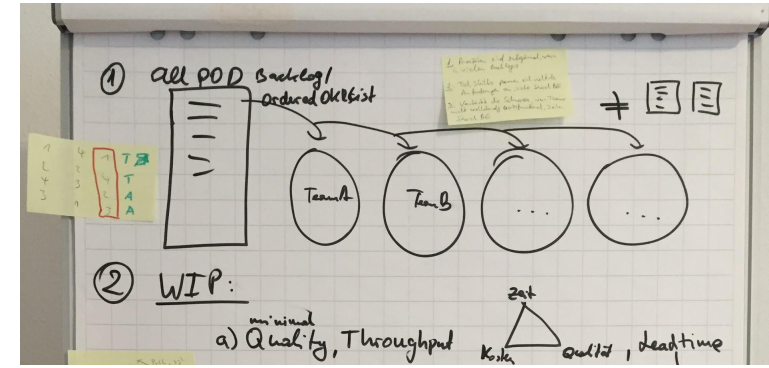
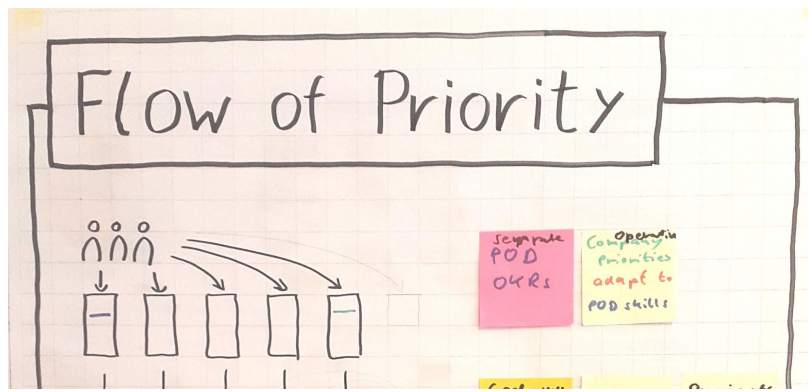
vs. continuous attention → less waste

→ What is tech debt: e.g. payment integration

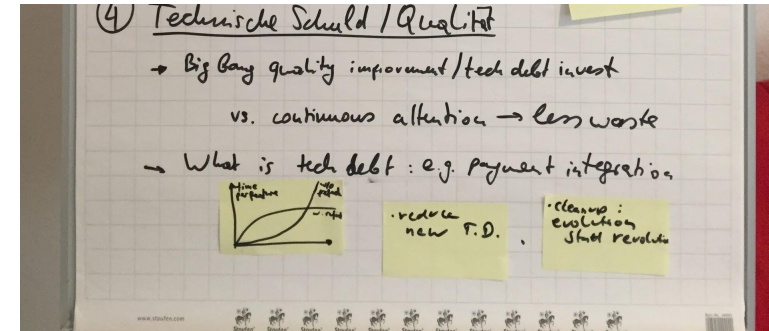
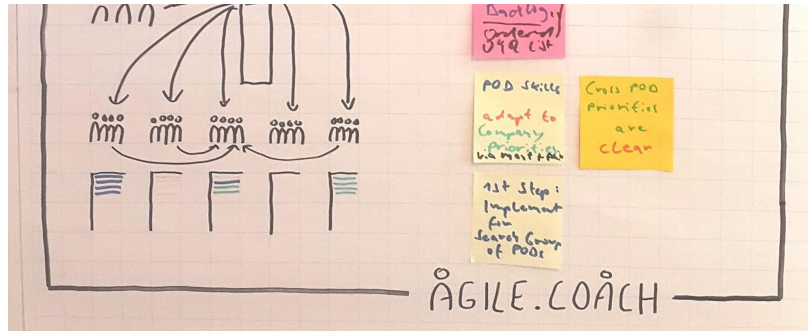


reduce  
new T.D.

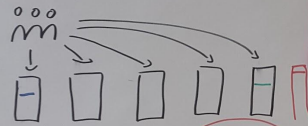
cleanup:  
evolution  
short resolve



# Vergessen im Büro

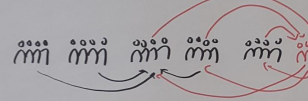


# Flow of Priority



Separate  
POD O/Rs

Actual  
company  
priorities  
adapt to  
POD S&C's



$C_n \cdot X + P_n \cdot J$   
are created  
for urgent  
goals

$C_n \cdot X + P_n \cdot J$   
are slowed  
down by  
dependencies

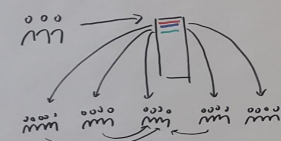
$C_n \cdot X + P_n \cdot J$   
slow down  
PODs



unclear  
(cross)  
priorities

risk of  
delay  
of urgent  
tasks

frequent  
escalation  
to top  
managers



"Ordered  
O/Rs list"  
"All POD  
Backlogs"

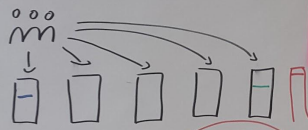
POD S&C's  
adapt to  
actual  
company  
priorities  
& dependencies



Clear  
(cross)  
priorities

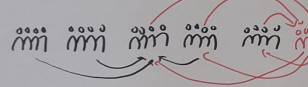
1st Step:  
Implement  
for  
several Comp  
of PODs

# Flow of Priority



Separate  
POD OKs

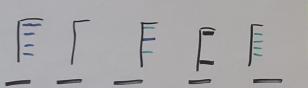
Actual  
company  
priorities  
adapt to  
POD Skills



Can X + Proj.  
are created  
for urgent  
goals

Can X + Proj.  
are slowed  
down by  
dependencies

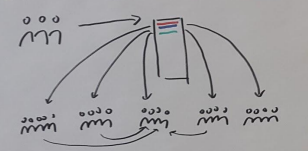
Can X + Proj.  
slow down  
PODs



unclear  
(cross)  
priorities

risk of  
delay  
of urgent  
tasks

Frequent  
escalation  
to top  
management



Ordered  
OKs list  
All POD  
Backlog

POD Skills  
adapt to  
work  
complexity  
from introduction  
to phase transfer



Clear  
(cross)  
priorities

1st Step:  
Implementation  
for  
separate group  
of PODs

## Work In Progress

| feature | value | duration | value/duration |
|---------|-------|----------|----------------|
| A       | 3k€   | 3 weeks  | 1              |
| B       | 7k€   | 4 weeks  | 1.75           |
| C       | 9k€   | 6 weeks  | 1.5            |

| Week        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | Cost of Delay     |
|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|-------------------|
| 1c together |   |   |   |   |   |   |   |   |   |    |    |    |    | 247 k€            |
| 8c one      |   |   |   |   |   |   |   |   |   |    |    |    |    | 157 k€            |
|             |   |   |   |   |   |   |   |   |   |    |    |    |    | Difference 90 k€! |

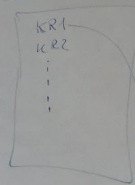


Finish the  
first, before  
starting  
new ones

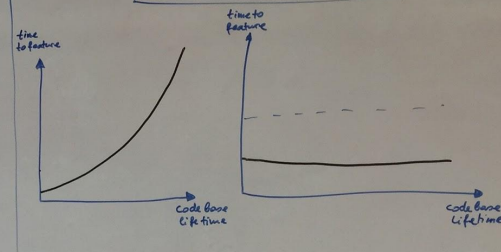
Measure  
items in  
Progress

Reduce  
WIP, measure  
start result

## ALL POD Backlog



## Code Quality



## Search Group

- AB
- AB
- BC
- AD
- CD

## Skills

| Team A | ABC |
|--------|-----|
| Team B | ABD |
| Team   |     |
|        |     |
|        |     |
|        |     |

ESKALATION AN CEO: "COACHES" SCHLAGEN EINE LÖSUNG VOR

# Reaktion CEO



- Abhängigkeiten zwischen Teams??
- Anspannung, Überforderung, Abwehr
- Selbst überlastet =>  
Leute sollen selbstständig auf Ziele hin arbeiten.
- Kein wahrnehmbarer Effekt des Gesprächs.  
Chance vertan.

ESKALATION AN CEO: "COACHES" SCHLAGEN EINE LÖSUNG VOR

# Learnings



- Beobachtungen anbieten
- Was will Coachee erreichen?
- ~~Lösung~~ -> Coaching / Methoden anbieten

# Danke für Eure Aufmerksamkeit! :-)

## SCRUMDAY 2019

**“Scrum ist ein Hilfsmittel die Welt zu verändern”**

**Wir hoffen, dass unsere Learnings euch dabei helfen,  
die Welt noch nachhaltiger zu verändern!**

Timon Fiddike & Sebastian Lange



@TimonFiddike & @agilebastian