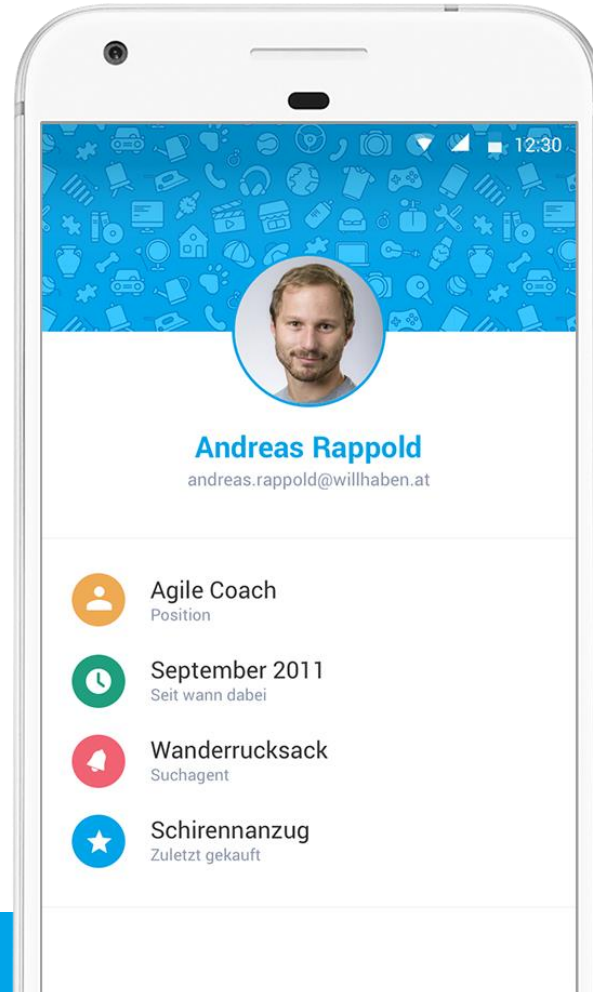


WILLHABEN

How we work



Hallo!



Kurzer Überblick über willhaben

willhaben

Stand 2018

- ca. 230 Mitarbeiter
- 70 in PD und PM
- 25 Java, 11 App, 12 Ops, 14 QA
- 8 PM

Stand 2011

- ca. 50 Mitarbeiter
- 9 PD-Mitarbeiter, 2 PM





Facts

5+ Mio. Ads

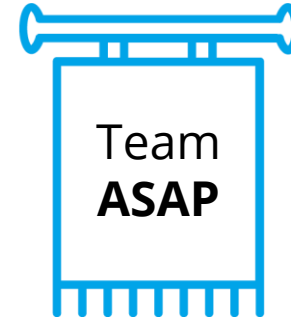
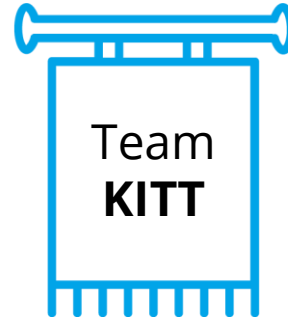
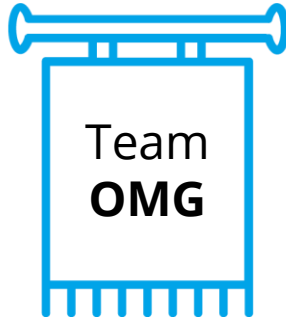
100.000+ New ads / day

30.000 Page impressions / min

3,35 Mio. Real people / month

Scrum Teams in willhaben

3 General Purpose Teams



Special Purpose Teams



Team
iOS-App Development



Team
Android-App Development

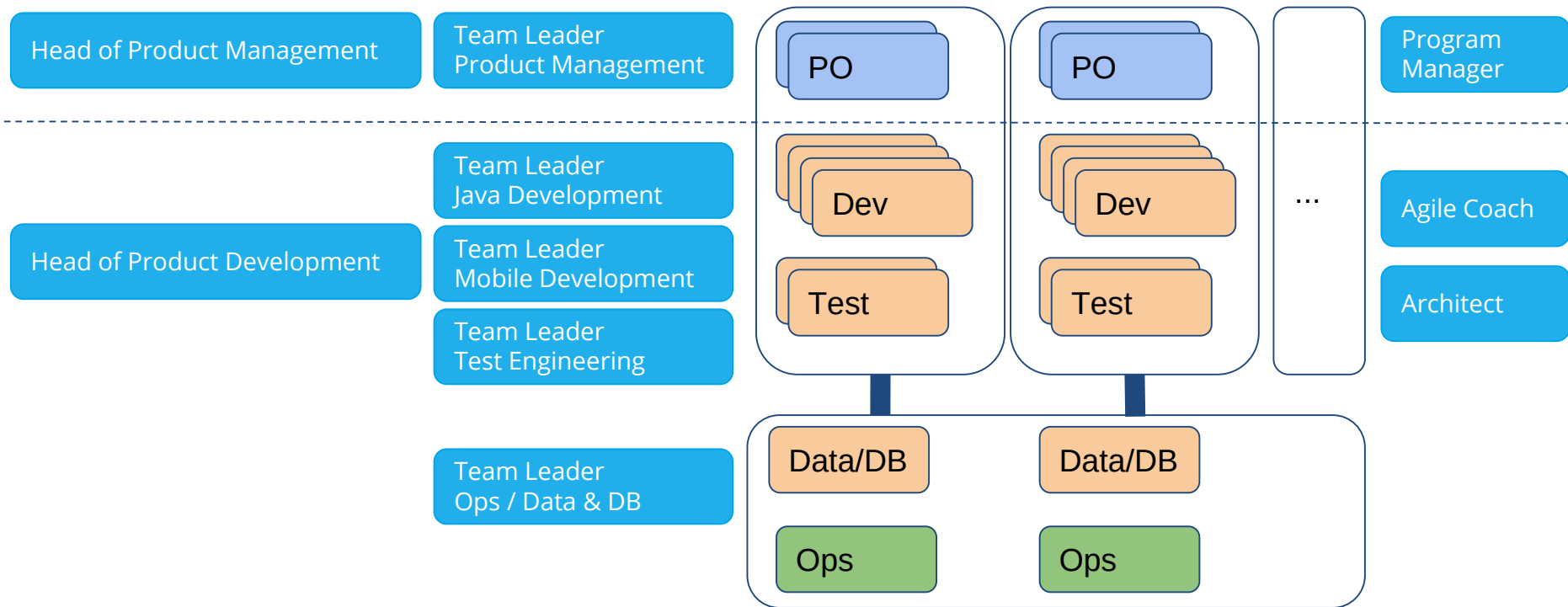


Team Job
betreuen die Standalone-
Job-Plattform von willhaben



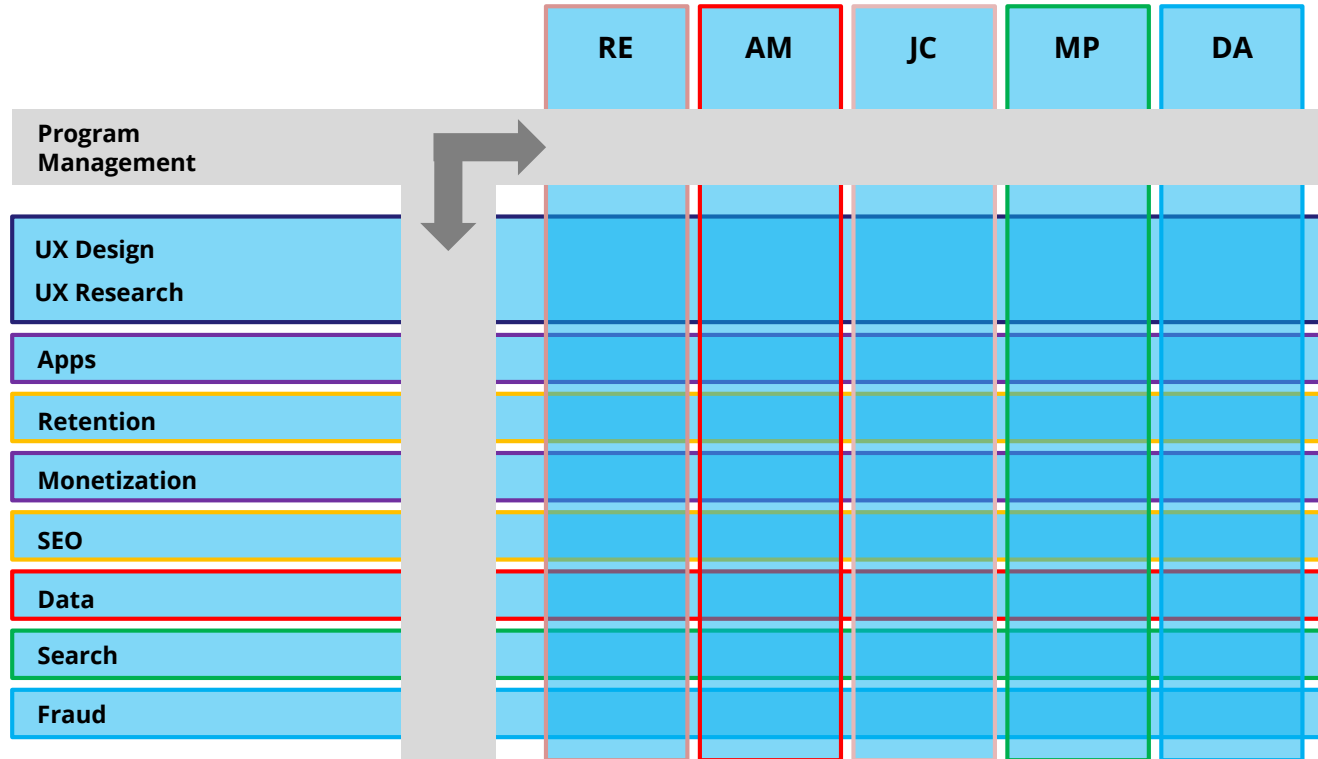
Team Fraud

Scrum Teams in willhaben: General Setup

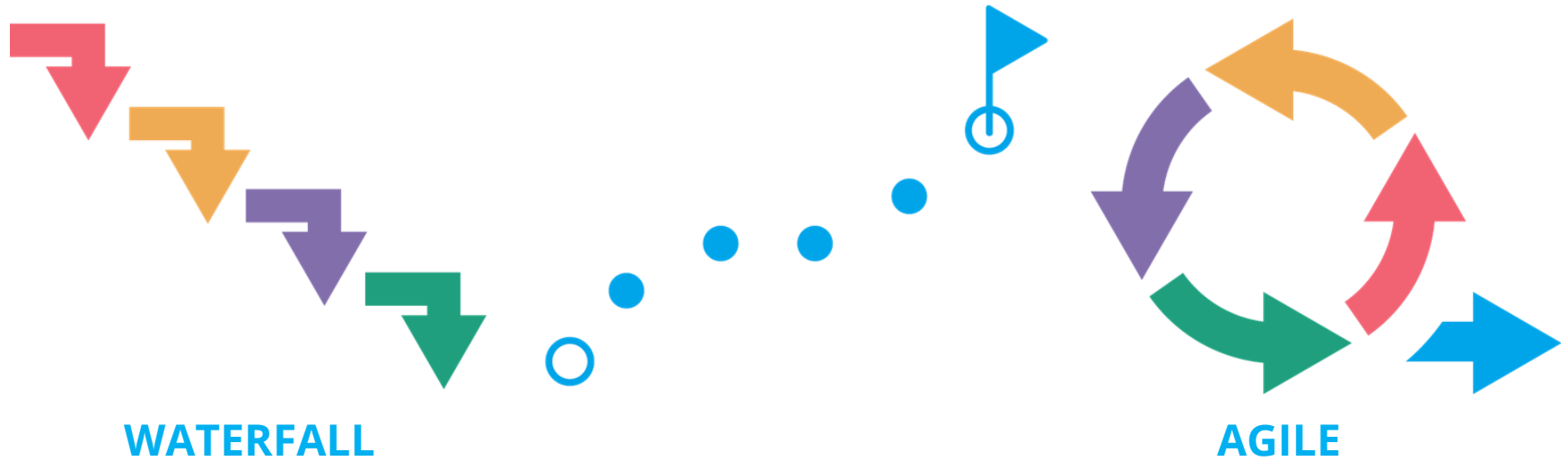


Wie wird die Arbeit in PM organisiert?

PM arbeitet als eine Matrix von cross-functional experts



Übergangsphase



How we work – Befragung

Hauptfragen

?

Warum herrscht eine Unzufriedenheit zwischen den zwei Abteilungen?

?

Was sind die größten Pain Points?

?

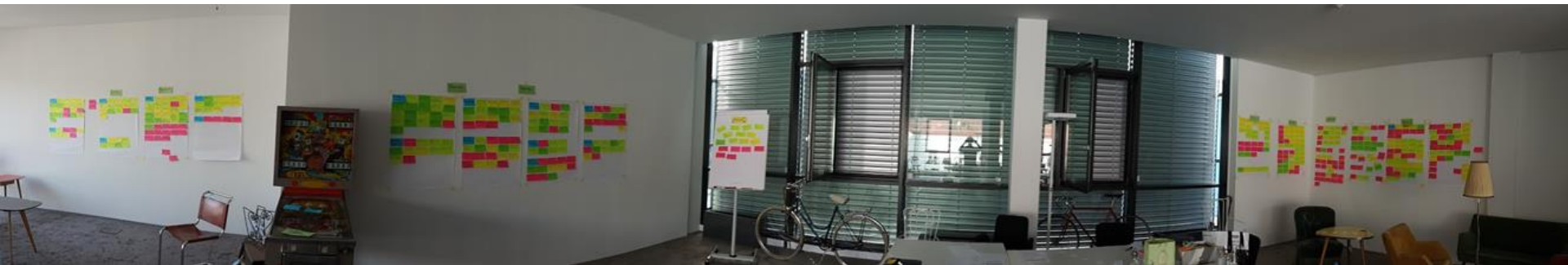
Was veranlasst die Mitarbeiter Frustration?

?

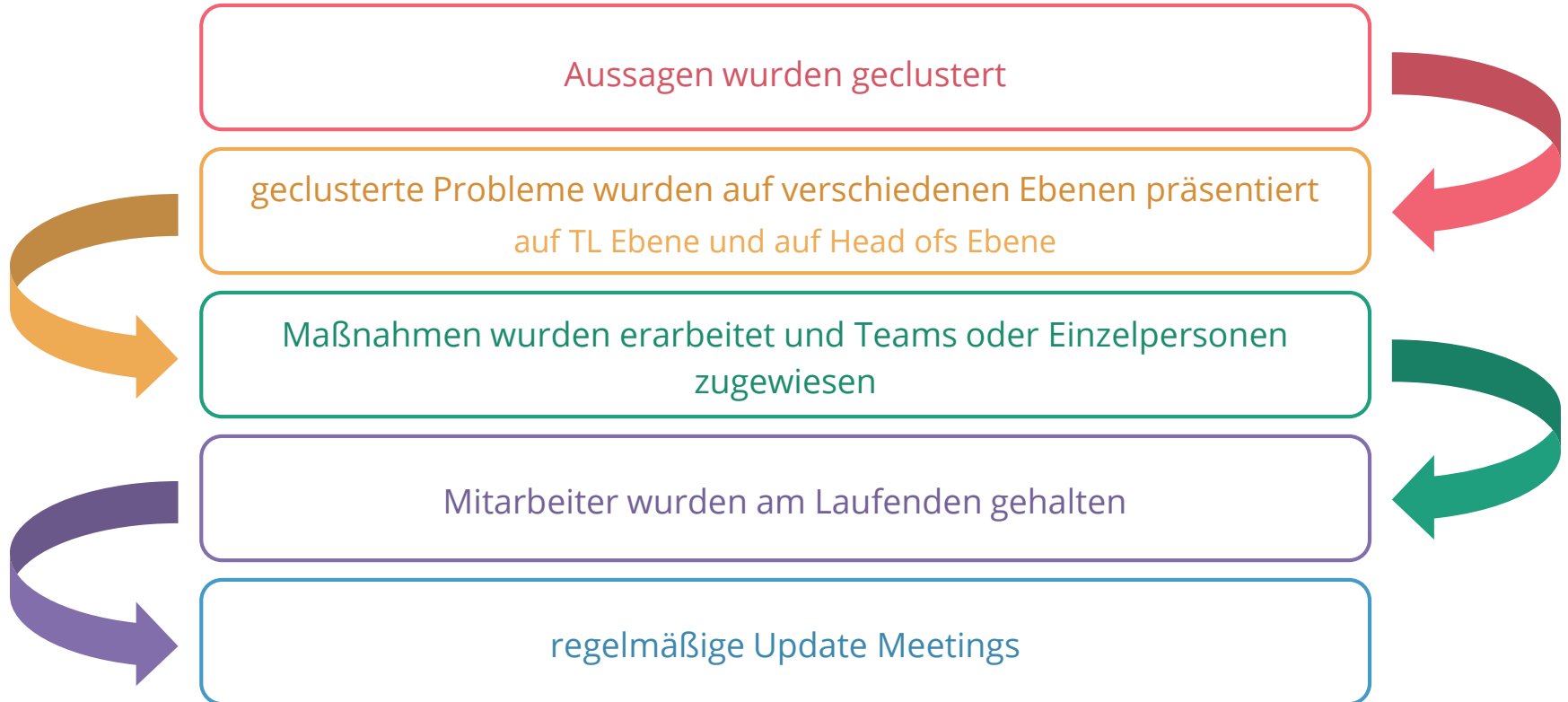
Wie können wir den Problemen begegnen und wieder mehr an einem Strang ziehen?

Durchführung

- zwei unterschiedliche Fragebögen für PM und PD
- Fragen zu den Bereichen Prioritäten, Produktmanagement, Arbeitsweise
- Feenfrage zum Schluss
- Crossover Befragung
- es wurden keine Teamleads befragt
- wichtigsten Statements wurden auf Post-Its festgehalten

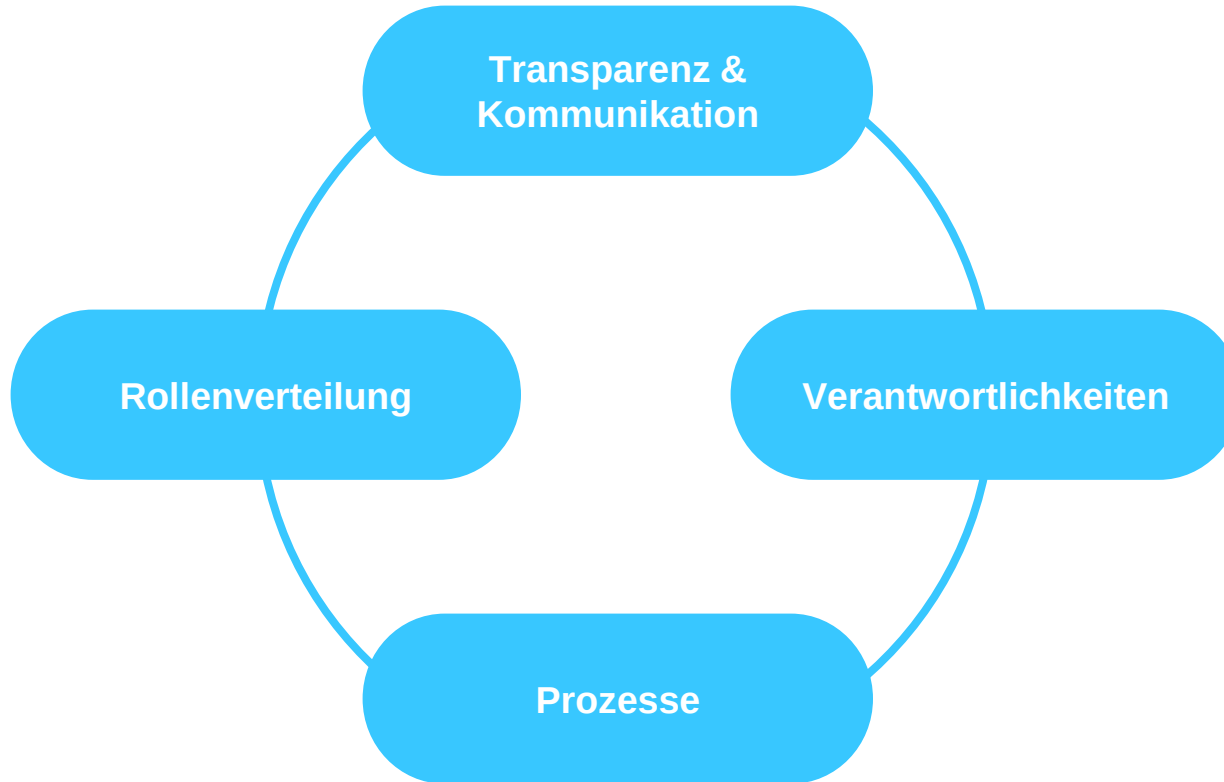


Weiteres Vorgehen



Die größten Pain Points

Pain Points geclustert in 4 Themengebiete



Pain Points

Transparenz und Kommunikation

Pain Points - Transparenz und Kommunikation



intransparente Projektverteilung auf Teams

fehlender Überblick über Projekte

Fehlendes Verständnis für das Warum und die Ziele eines Projekts der Devs

POs als unzureichende Motivatoren

POs fehlendes Vertrauen in technische Expertise der Devs

Warum und Ziele



bessere und detaillierte Kommunikation



neue Modi und Formate finden (Roadmap Update, Kickoffs,...)



Ziele müssen allen Projektbeteiligten klar sein



Wiederholen des Warums da es im Laufe des Projekts verloren geht



Out of scope



Monatliches Roadmap Update



Kickoffs, Projektmeetings, Workshops

Outcomes und Userfeedback



nach der Release den Informationsfluss aufrecht halten



Was sagen die User?



Wie verändern sich die KPIs?



Haben wir die Projektziele erreicht?



Teil des Roadmap Updates (für alle QA, Ops, UX, Devs, PMs)



Sprintplannings

Pain Point Rollenverteilung

Pain Points - Rollenverteilung



Zuständigkeiten der POs nicht allen Devs klar

Unterschiedliche Bilder was ein Dev von einem PM erwartet und vice versa

Unterschiedliche Ansichten von PM und PD für die Rolle des POs

Fehlender Ansprechpartner bei den Devs für den PO

Technischer Projektleiter

- Rolle besser definieren
 - TPLs an ihre Verantwortung erinnern
 - für alle größeren Projekte einen TPL einsetzen
-
- ➔ Erfolg hängt sehr an den Skills des TPLs
 - ➔ temporäre Rolle
 - ➔ Ownership liegt im Team nicht bei einer Person



Pain Point Verantwortung

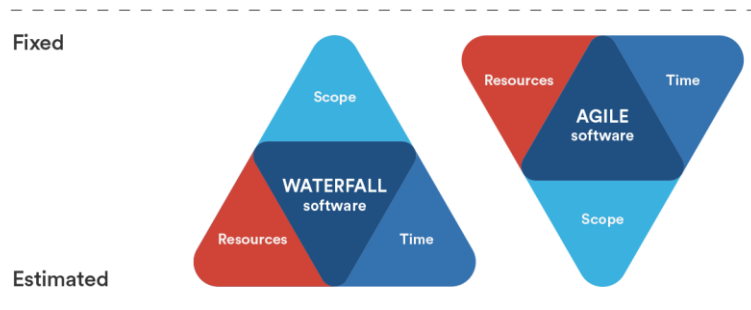
Pain Points - Verantwortung



terminliche Verschiebungen sind egal

Verbindlichkeit

- Termintreue ist wichtig
- Terminverschiebungen nur mit guter Begründung
- PM muss Verschiebungen hinterfragen
- PM/TPL finden gemeinsam eine Lösung (Scope vs Date)
- Commitment des gesamten Teams



<http://lukeangel.co/project-management/whats-a-workback-schedule/attachment/waterfall-v-agile-iron-triangle-v03/>

Pain Point Prozesse

Pain Points - Prozesse



Innovationen kommen nicht in Produktion

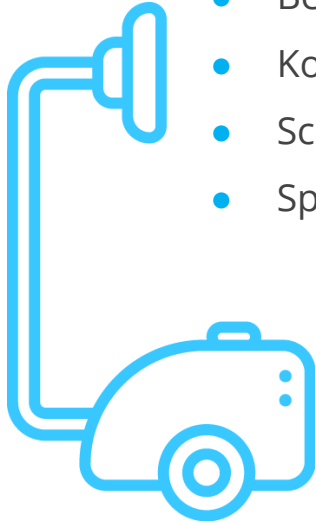
Wir bauen zu viel technische Schuld auf

Zu lange Meetings

Frontend Teams sind schlecht mit Backend Teams abgestimmt

Housekeeping / Bug Workflow / Schätzungen/...

- Inhärentes und dezidiertes Housekeeping
- Bewusstsein für Qualität bei PMs schaffen
- Komponentenownership stärken
- Schätzen je Team unterschiedlich
- Sprintziele und Sprintdemos



Futuregarden & Crazy Friday



- viel Frust bei PD
- wollen Resultate in Produktion sehen



- TechDay/Futurgarden: 1mal im Quartal
- InnovationDay (Shark Tank) – 2mal im Jahr

Apps

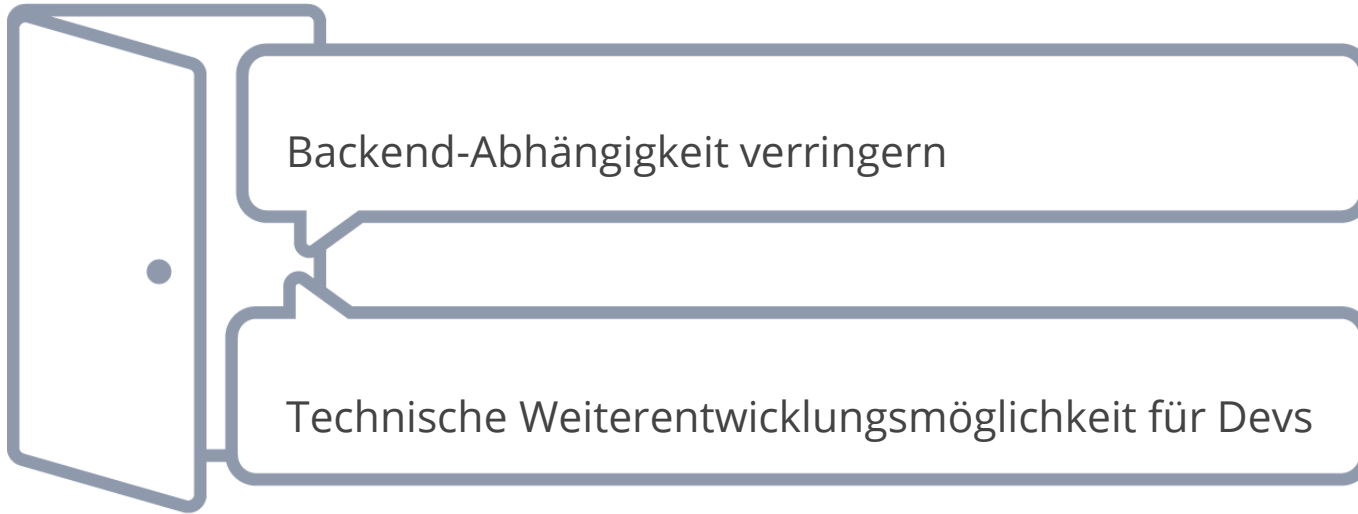


- Team ist sehr/zu groß
- Standups/Plannings lang und “unproduktiv”
- Jira ist unübersichtlich



- ☀ Trennung in Android und iOS
- ☀ technische Qualität und Stabilität ist wichtig
- ☀ Teams müssen nicht gleich auf sein

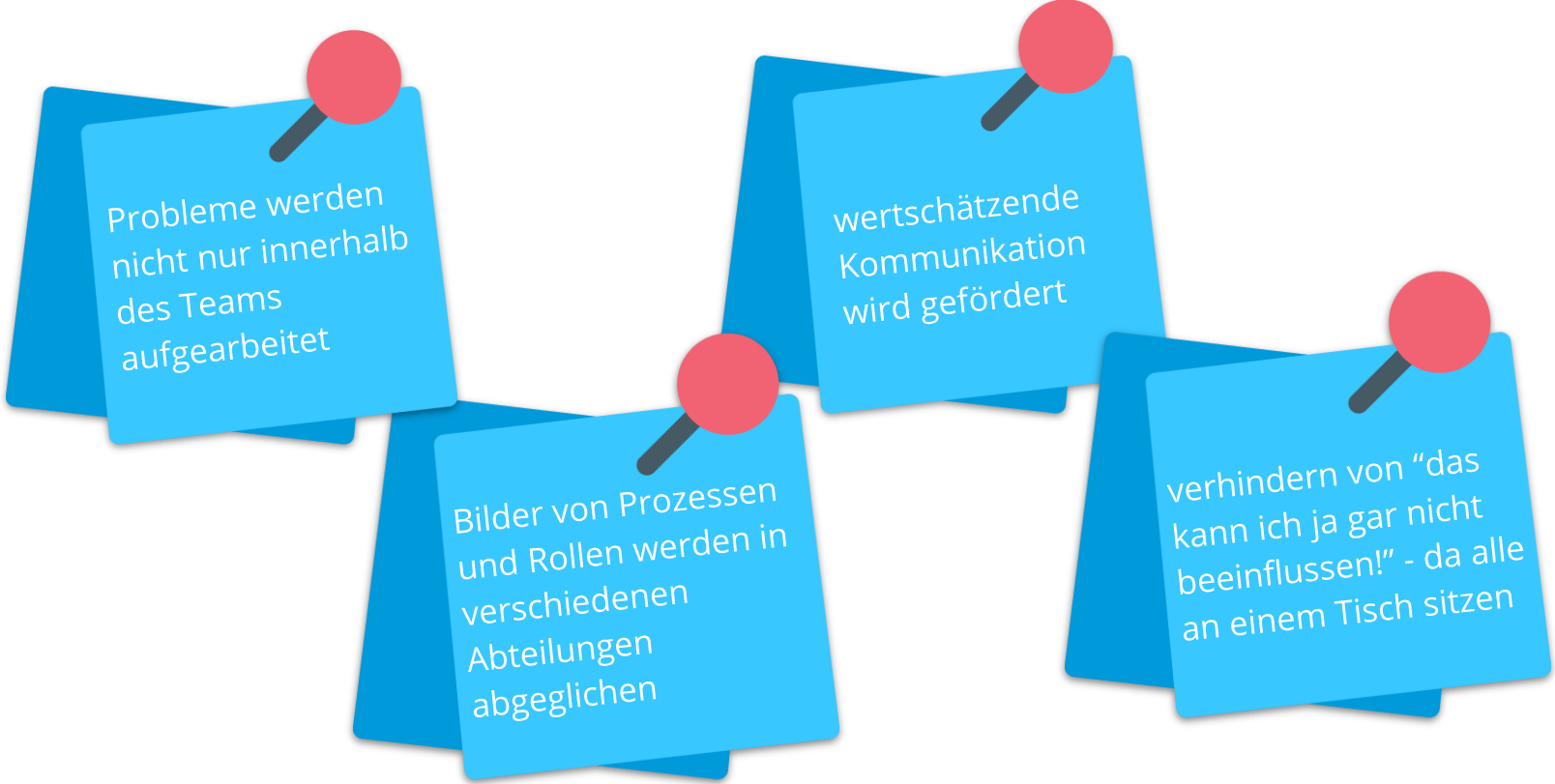
Backend Know How bei Apps



- ✓ erste Einschulungen haben stattgefunden
- ✓ zwei App Entwickler sind bereits soweit auch komplexere Features von Backend bis Frontend zu entwickeln

... und ...

Abteilungsübergreifende Retrospektiven



Probleme werden
nicht nur innerhalb
des Teams
aufgearbeitet

wertschätzende
Kommunikation
wird gefördert

Bilder von Prozessen
und Rollen werden in
verschiedenen
Abteilungen
abgeglichen

verhindern von "das
kann ich ja gar nicht
beeinflussen!" - da alle
an einem Tisch sitzen

Danke für die Aufmerksamkeit!