

7 Dinge wie man gute Teams großartig macht ①
Svenpet.com (Durchschnittsalter der Teams ~ 35 J)

"Be the change you seek"

1) It's Flowtime

Flow vs Kollaborieren

↳ "Do not disturb" - time: Zeit festlegen (z.B. 13-16h)
Kommunikation außerhalb des Büro

↳ 1. Woche: keine weder Störungen + Erklärungen

↳ "Do not disturb the rest"

↳ Eines (jeweils) für Support ansprechbar ("Support hat der Tag")

↳ wechselt

↳ Voraussetzung: jeder kennt den Code ausreichend

↳ Supporter beinhalten (T-Shirt, Kape...)

↳ Reality check

↳ Machbarkeit 4 von 5

↳ Großartigkeit 2 von 5

2) Feed your brain

↳ organisierte coding sessions, z.B. halbes Jahr lang 1 Abend pro Woche

↳ z.B. um neue Programmiersprache zu lernen

↳ es können u.U. nicht ~~weder~~ teilnehmen

↳ in berichten während des Mittagessens (Vortrag)

↳ schwierig, mit 8 Leuten 1 Vortrag pro Woche organisieren

↳ alternativ: 1x pro Woche Vortrag gemeinsam sehen (Web)

↳ reality check: Machbarkeit 4 von 5
Großartigkeit 2 von 5

* des Teams durch diese Maßnahme

7 Dinge...

(2)

3) Auch kleine Dinge wertschätzen

↳ Anerkennung ist wichtig für MA-Zufriedenheit

↳ Anerkennung öffentlich ~~machen~~ machen

↳ Anerkennung für jeden möglich (jeder kann anerkennen) nicht
nur Chef

↳ KUDOS

↳ Lob-Tickets eintragen (z.B. Thomas hat x
gut gemacht + trinkt gerne Rotwein)

=> Klicks, Feeding + Frische Wein

↳ Real. durch 2 von 5
3 von 5

4) Report Robot ("human report robot"). Daten, die gesammelt
werden: Zeit,

Features,

Codezeilen, ger reviewte und nicht

Builds,

Buildzeit

Testserver Performance,

Velocity,

Lead Time,

Verkäufe,

Supportfälle,

Antwortzeiten,

Bereisungen, ...

→ Automatische
Reports

7 Dinge...

(3)

Infoscreens, mind. 1 pro Team mit Burndownchart,
Builds (fließende + fehlgeschlagene), kein abtiner, Nachschlagen
notwendig

- ↳ Real-check 2 von 5 Machbarkeitspunkte
4 von 5 Großartigkeitspunkte
↳ da kurze Feedbackzyklen

5) "Eat your own dog food"

↳ Alpha tests

↳ alle aus dem Unternehmen können testen

↳ Kunden besser verstehen

↳ SW selbst ausprobieren, um Anwender Schmerzen selbst
zu fühlen (besser, nicht optimal, nicht absolut)

↳ Weg, um kurze Feedbackzyklen zu haben

↳ macht Testen effizienter

~~100~~

↳ Reality check

2 von 5 Machbarkeit

5 von 5 Großartigkeit

6) Experimentation Time

• Slipt-Days: Prototyp, innerhalb von 24 Std anliefern

↳ Innovation ←
↳ Motivation durch

• 1x pro Quartal

7 Dingen...

(4)

- Ideen aufschreiben, um die besten Ideen bilden sich Teams
- Plan machen, Mini-Sprint-Planung
- Ab dann: 24 Std lang codieren (Do 15 Uhr: "Go!")
16 Report Energy Drinks

↳ Code Convention, Designs etc. jetzt unwichtig, es muss nur laufen (außen)

↳ Fr: je 5 Min (pro Team) Ergebnis präsentieren

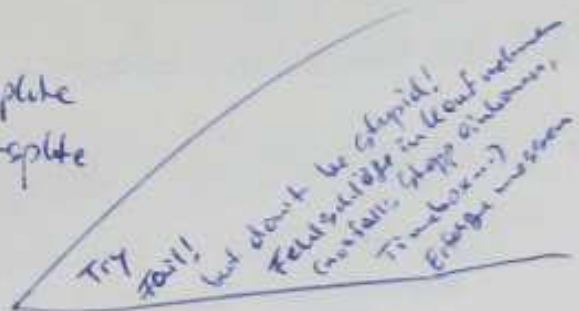
↳ Gewinner: Idee mit dem größten Wow-Effekt

↳ alle sind anwesend bei Präsentation (z.B. auch Marketing)

↳ => Happy Nerds

- Real-check 4 von 5 Machbarkeitspunkte
5 von 5 Großartigkeitspunkte

7) 20%-Zeit für eigene Ideen



↳ z.B. auch für Code-Refaktoring (außen)

- nur für eigene Produkte zu verwenden (nicht für Fertigkeit)

• Bei mehr als 40 Std Aufwand muss man 3 Supporter im Unternehmen suchen, die das Projekt für sinnvoll halten

• Bei mehr als 80 Std Aufwand ist das OK von oben nötig

• Problem: Iteration erfordert mehr Aufwand, man möchte das Team nicht im Stich lassen

↳ Hilfe: statt 1 Tag/Week regelmäßig "Innovationweek" einrichten

- Real ch: 4 von 5 Machbarkeit
5 von 5 Großartigkeit (steepet.com/slides)