14:00 – 14:45

15:00 – 15:45

16:00 – 16:45

Coaching

303 Kleiner Saal

Battle „Papier vs. Software"

#noestimates

Large-Scale Scrum

302 Gesellenzimmer (Konferenzraum)

3 → Fizz
5 → Buzz

Informations-häppchen

Funktionale Mikroservices

204 Baulhüttensaal

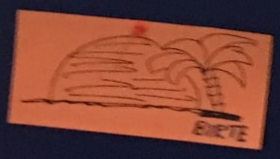
Verantwortung in Teams?

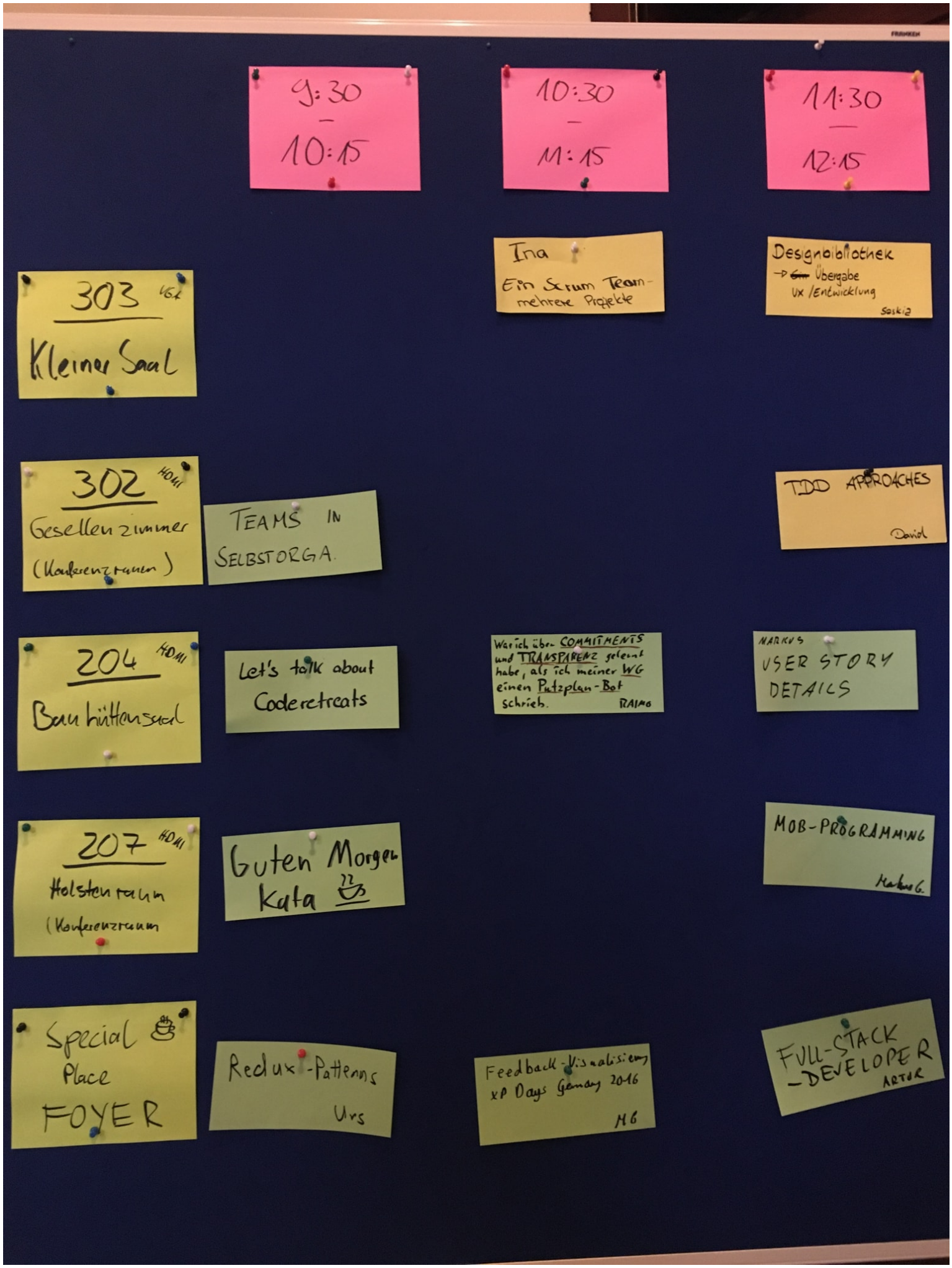
Agile Transition

207 Hütten-Raum (Konferenzraum)

Lernen lernen
↳ Spezialist sein & stetig neues Lernen

Special Place FOYER




9:30 – 10:15
10:30 – 11:15
11:30 – 12:15
303 VGA Kleiner Saal
Ina Ein Scrum Team - mehrere Projekte
Designbibliothek → ~~Grün~~ Übergabe UX / Entwicklung Saskia
302 HDMI Gesellenzimmer (Konferenzraum)
TEAMS IN SELBSTORGA.
TDD APPROACHES David
204 HDMI Bauhüttensaal
Let's talk about Coderetreats
War ich über COMMITMENTS und TRANSPARENZ gelernt habe, als ich meiner WG einen Putzplan-Bot schrieb. RAIMO
MARKUS USER STORY DETAILS
207 HDMI Holstenraum (Konferenzraum)
Guten Morgen Kata
MOB-PROGRAMMING Markus G.
Special Place FOYER
Redux-Patterns Urs
Feedback-Visualisierung XP Days Germany 2016 MG
FULL-STACK-DEVELOPER ARTUR

18

Software

01001101

Revisionierbarkeit.
+ Kommunikation dokumentiert
Informationen wiederauffindbar

+ Informationen stets zugänglich
+ durchsuchbar

+ Dokumentation:
Jira-Tickets
mit Commits
↳ PO kann das sehen

automatisierte Akzeptanztests würden das auch können

− Unübersichtlich (Bildschirm zu klein)

+ Attachements
− verleitet zu Overengineering

− Upgrade-Horror

− Ressourcenverbrauch (Strom, CO2)

− asymmetrische Kommunikation in Gruppen-Settings

− zu starker Fokus auf Reporting

+ Reporting „automatisch"
Datenerfassung „

− Benachrichtigungen nerven

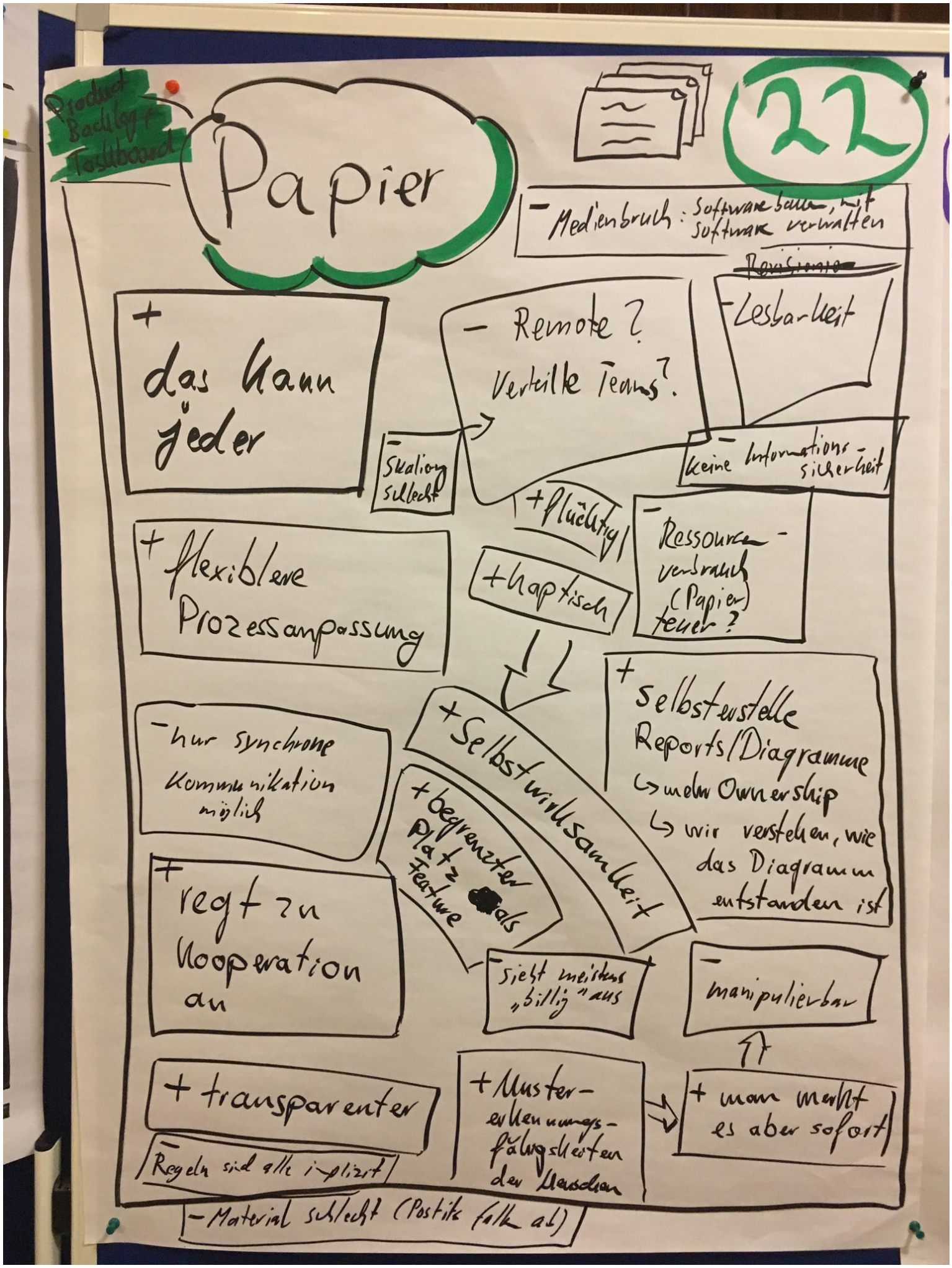
− braucht Schulung

− versteht jemand, wie die Reports zustande kamen

+ Unbegrenzter Platz

− Zugangshürde

− vollgemüllt, ohne es zu merken ← aufräume automatisierbar


Product Backlog + Taskboard
Papier
22
- Medienbruch: Software bauen, mit Software verwalten
Revisionen
- Lesbarkeit
+ das kann jeder
- Remote? Verteilte Teams?
- Skaling schlecht
- keine Informations-sicherheit
+ flexiblere Prozessanpassung
+ flüchtig
+ haptisch
- Ressourcen-verbrauch (Papier) teuer?
+ Selbstwirksamkeit
+ selbsterstellte Reports/Diagramme
↳ mehr Ownership
↳ wir verstehen, wie das Diagramm entstanden ist
- nur synchrone Kommunikation möglich
+ begrenzter Platz als Feature
+ regt zu Kooperation an
- sieht meistens „billig" aus
- manipulierbar
+ transparenter
Regeln sind alle implizit
+ Muster-erkennungs-fähigkeiten der Menschen
+ man merkt es aber sofort
- Material schlecht (Post-its fallen ab)

Verantwortung

Mitdenken

tu was gemacht werden muss um den Job fertig zu machen

dafür ----- Fixen wenn kaputt

sorgen das es wieder gut wird

den großen Rahmen sehen

Produkt & Technologie

klarer Rahmen für was Verantwortung übernommen werden soll

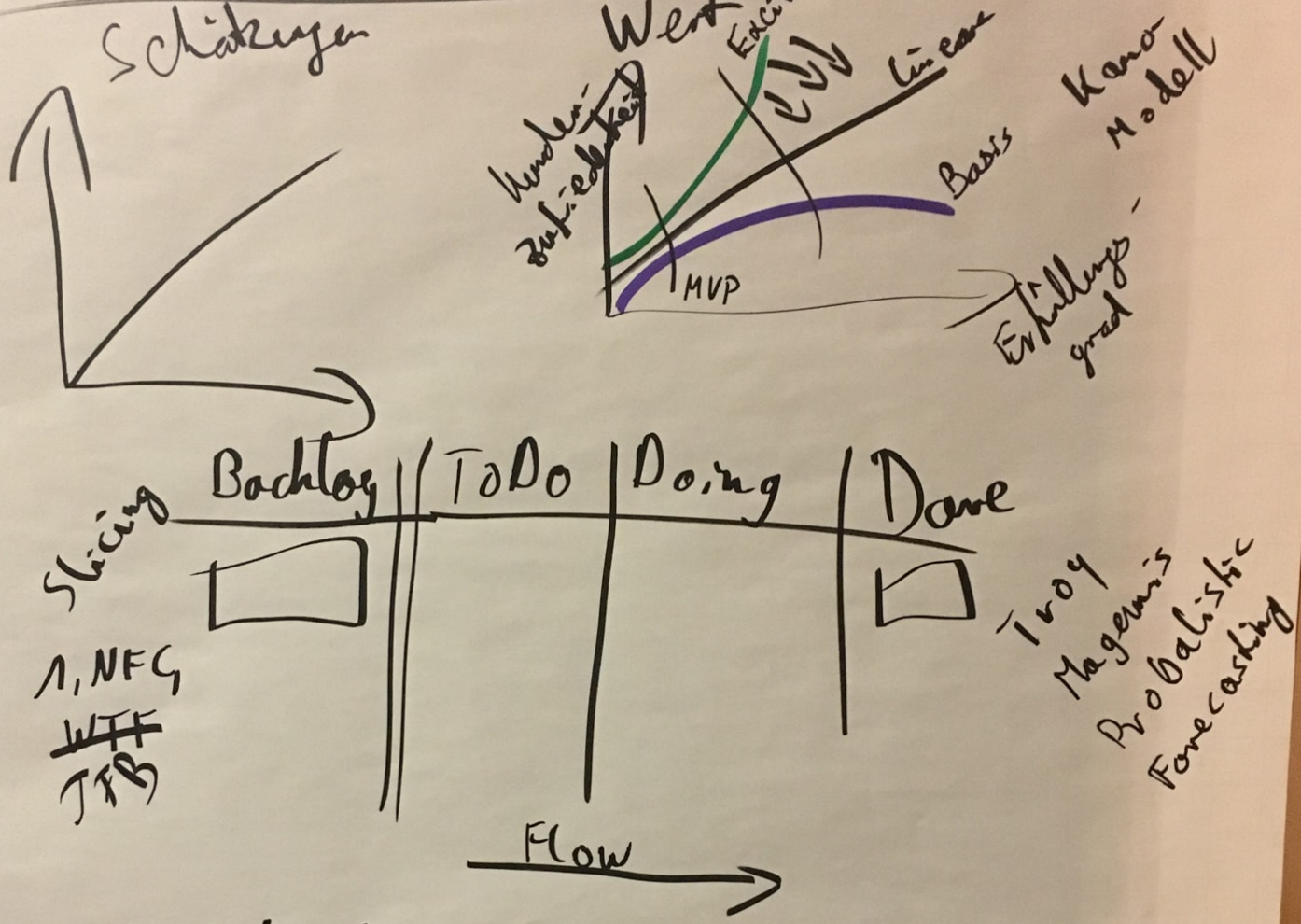
Kommunikation & Transparenz

Lernen

Bewusstsein für Verantwortungsübernahme

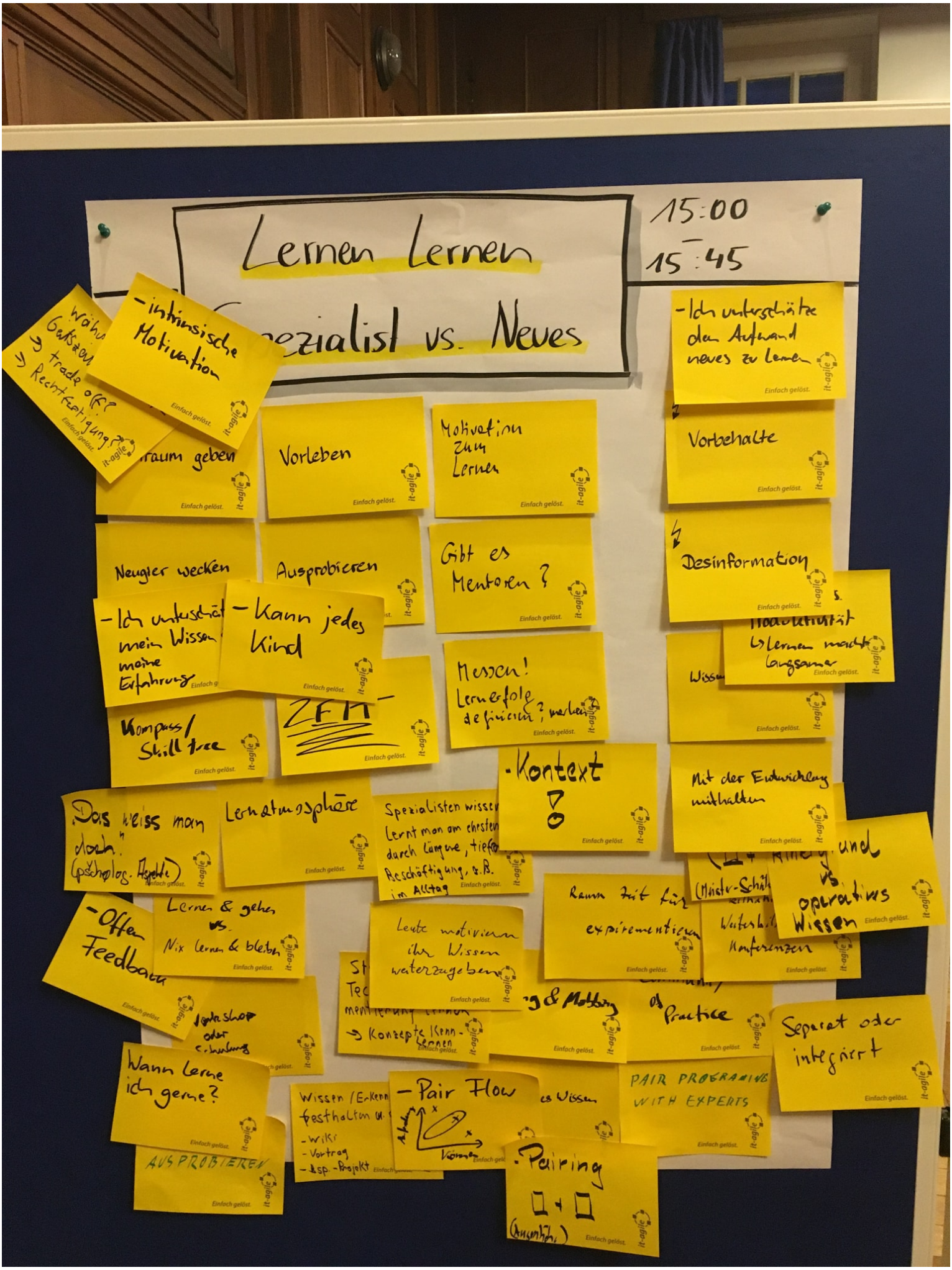
Fehlerkultur in Unternehmen hilft Verantwortung zu übernehmen

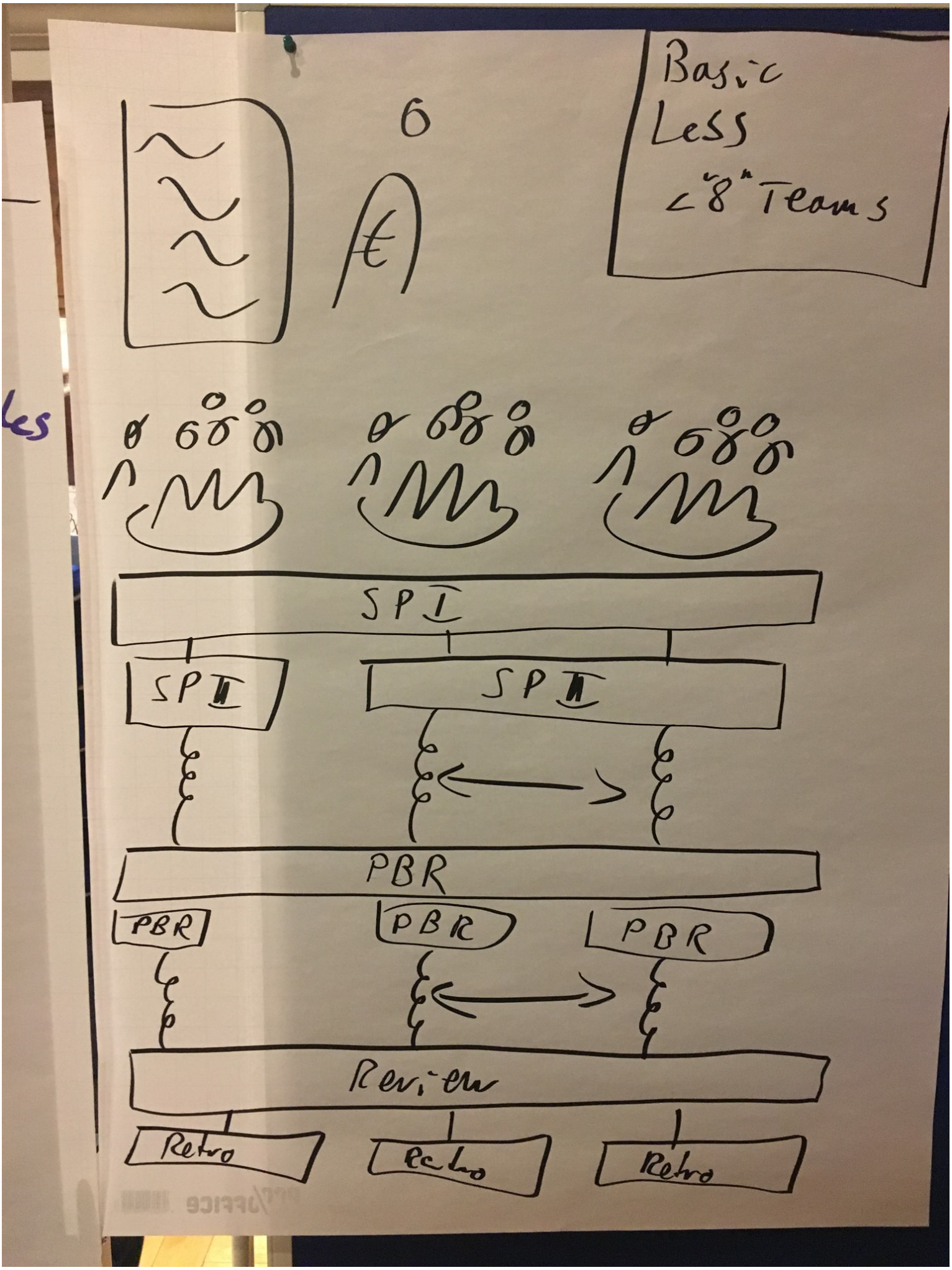
#noestimates



Entscheidungen, die heute auf Schätzungen basieren:

- Koordination zw. Teams
- Aufgabenklärung
- Forecasting
- Priorisierung

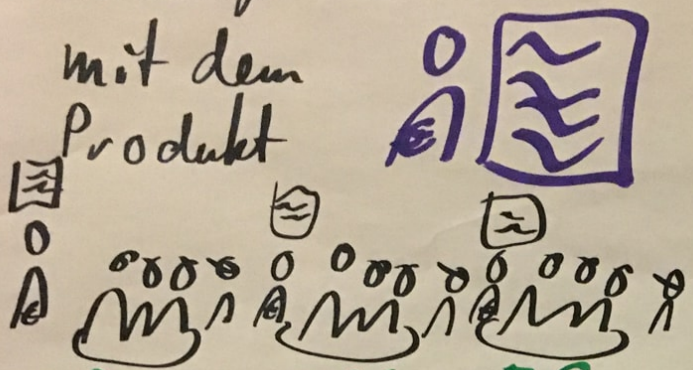

Lernen Lernen
Spezialist vs. Neues
15:00
15:45
- intrinsische Motivation
- Ich unterschätze den Aufwand neues zu lernen
Vorleben
Motivation zum Lernen
Vorbehalte
Neugier wecken
Ausprobieren
Gibt es Mentoren?
Desinformation
- Kann jedes Kind
Messen! Lernerfolg definieren? messen
Kompass / Skill tree
ZEIT
- Kontext
Mit der Entwicklung mithalten
Das weiss man doch. (pschyolog. Aspekte)
Lernatmosphäre
Lernen & gehen vs. Nix lernen & bleiben
- Offen Feedback
Leute motivieren ihr Wissen weiterzugeben
Kann Zeit für experimentieren
operatives Wissen
Practice
Separat oder integriert
Wann lerne ich gerne?
- Pair Flow
PAIR PROGRAMMING WITH EXPERTS
AUSPROBIEREN
- Pairing
Einfach gelöst.
it-agile


Basic
Less
< "8" Teams
SPI
SPII
SPII
PBR
PBR
PBR
PBR
Reviews
Retro
Retro
Retro

Large-Scale Scrum (LeSS)

http://less.works

Identifikation mit dem Produkt



Lernen

crossfunktionales Lernen

Mitgestalten / Einfluss auf Produkt

Product Backlog Refinement

@vradczenski

Transparenz, Commitments & Putzpläne

▷ Das Problem

- Putzräder kennen keine Zeit und haben kein Gedächtnis 
- Ego's "die eh immer alles machen"

▷ Die Lösungsansätze

- Gamification durch Sternchen
- Commitments verfallen nicht
- Werteradar

▷ Der Chatbot

- 3 Hero Tasks (Küche, Flur, Bad) -> 5 Punkte, fest zugewiesen
- Adhoc erledigte Aufgaben "/erledigt Pfand weggebracht" -> Voting von 1-3 Punkten von allen

▷ Die nächste Version

- Challenges, für die man kollektiv eine Belohnung festlegt. "/challenge Fenster putzen"

Agile Transition

Wenig Raum für Kreativität

Einfach gelöst.

Kunden-nähe

Einfach gelöst.

Mitgenommen
→ Manager haben auch Probleme → Warum?
→ Lösung muss nicht agil sein
→ Immer die gleiche Problem aber neu benennen
→ Agile Lösung für das falsche Problem?
→ Echt schweres/großes Problem 5/10 Jahre

Fremdsteuerung VS.

6

→ Nur von oben funktioniert nicht
→ Handwerkszeug für Mitarbeiter
→ Für jede Ebene den passenden Coach
→ Führungskräfte arbeiten im Team agil
→ Kontrolle abgeben
→ Warum auf Fk-Ebene beantworten

→ Grundlegender Werte- und Kulturwandel
→ Beispiel: Otto-Shop
↳ Fach- & IT-Seite gemeinsam
→ Glaube an Teams
→ Agiles Council vs. Agiler Papst
→ Menschen sind unterschiedlich

pers. Entwicklung

Einfach gelöst.

Pull

Ökonomische Motivation

Einfach gelöst.

Transparenz macht angreifbar (Politik)

Einfach gelöst.

Unterschiedliche Bewegungen in eine gemeinsame

Estimation-Points
≠ Zeiten
≠ fix

Einfach gelöst.

3

Begleitet von

Agile Coach etablieren!

Einfach gelöst.

Was mache ich mit "Teamleiter" meinen fachlich und/ =

→ Aufhören Produkte zu verkaufen
→ Mittlere Führungsebene hat Angst "wegoptimiert" zu werden

Dumm-halten

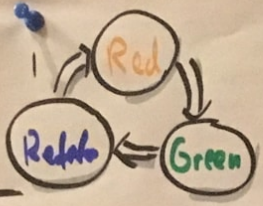
Wissensaustausch zwischen Teams

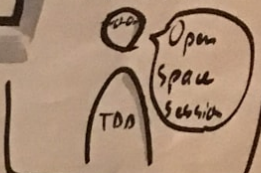
Einfach gelöst.

Einfach gelöst.

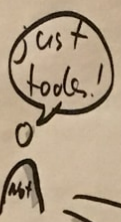
TDD Approaches

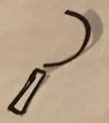
Von:
David Völkel





Classicist vs. Mockist





- Mocks eher an Systemgrenze
- bottom-up oder emergentes design

- kollaboration testen
- isoliert testen
- outside-in design

Trade-offs? Kombination?

Unsicherheit

Wieviel weiß ich über Struktur?

Emergent Design

- Bei unklaren Strukturen
- Aufwändig

⇒ Bei wenig Wissen

⇒ NICHT bei bestehender Struktur

Treiber

ORTHOGONAL ZU MOCKS!

Inside-Out

- Adapter direkt
- du

⇒ Adapter isoliert
⇒ Fixes Driftsystem

Outside-In

- entweder Mocks
- oder Fake-It

⇒ Struktur folgt Use Case

Isolation

Aufwand

ZU GROSS
- Speed ⇒
- Fehlerfindung

vs

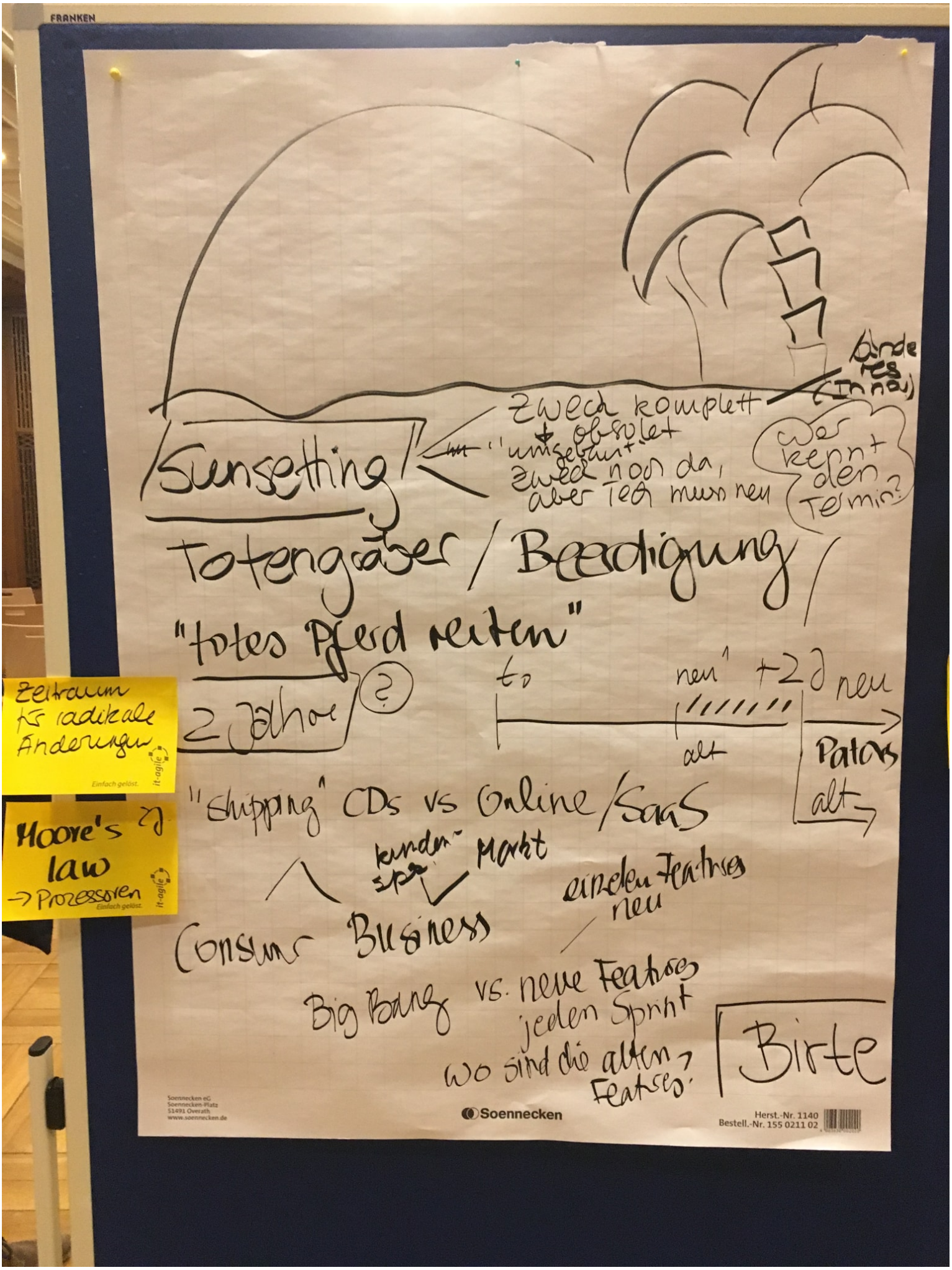
zu klein
- Lesbarkeit
- Low Refactorability

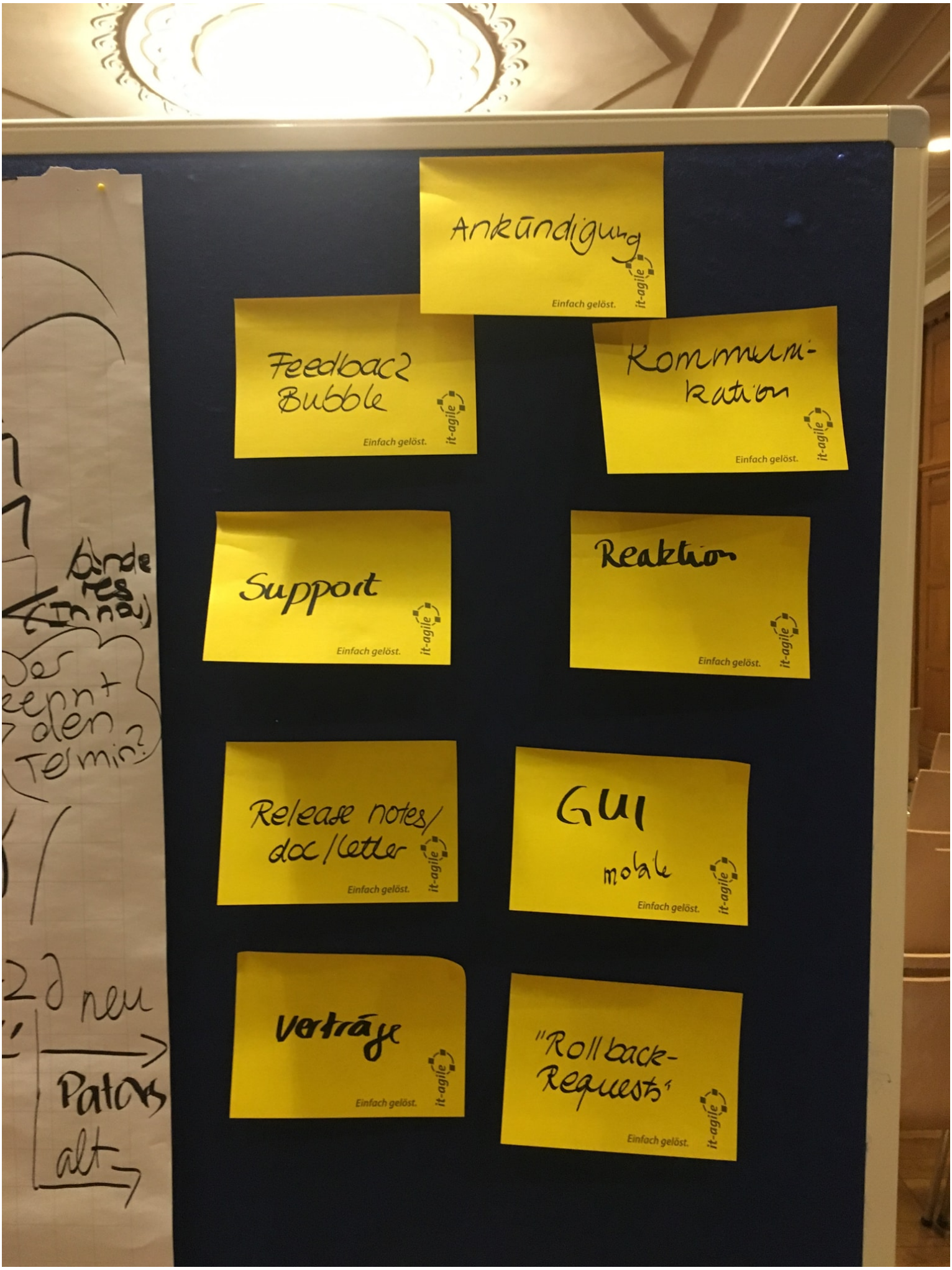
Algo:

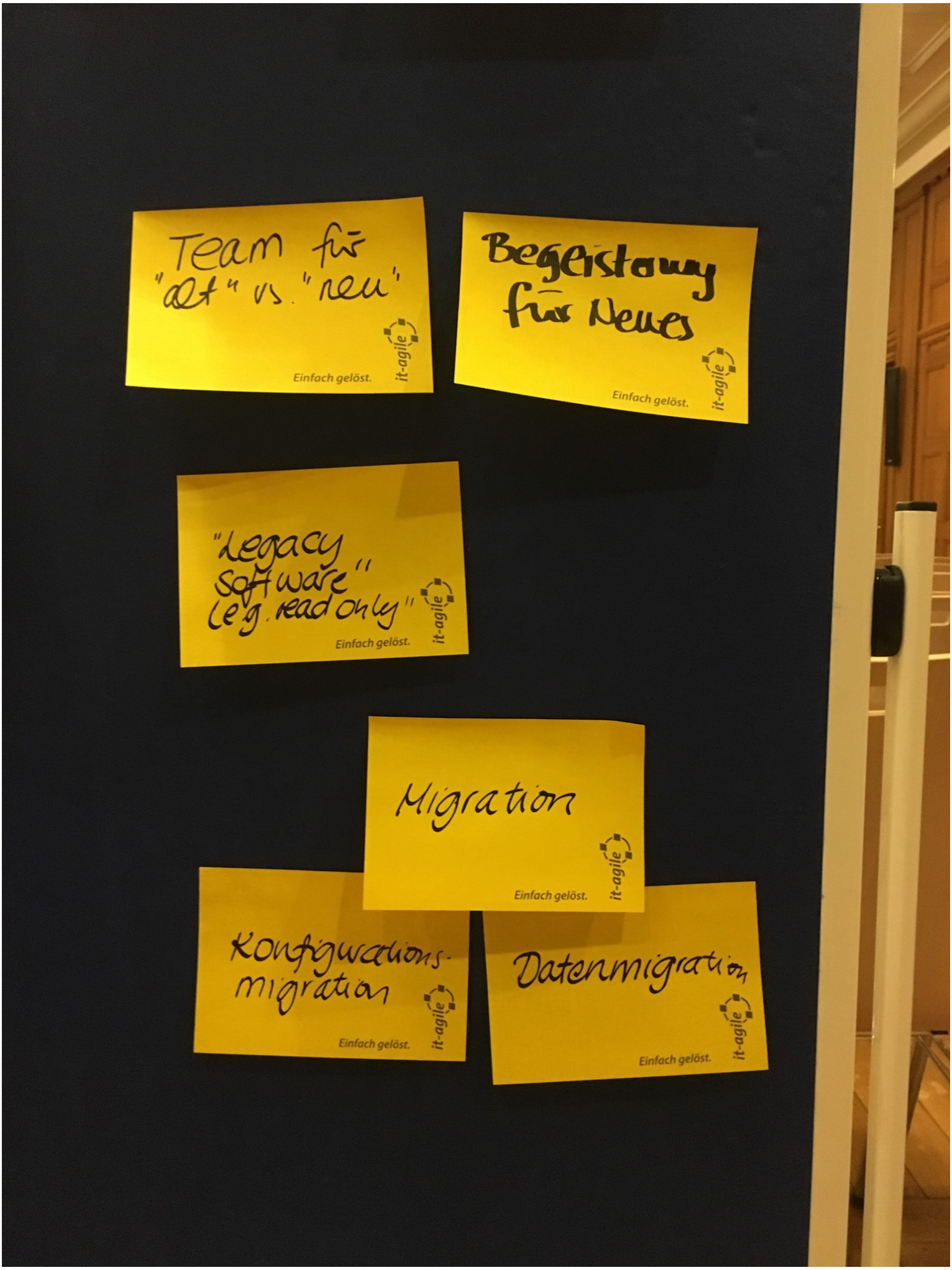
if unsicher: Emergent
else if driftsys: Inside-Out
else [illegible]
else: Outside-in
Fake-It

USER STORY CREATION

- HÄNGT VON DOMAINWISSEN AB
- VORGEGEBENE LÖSUNGEN UNTERDRÜCKEN DISKUSSIONEN / KREATIVITÄT


FRANKEN
Sunsetting
Totengräber / Beerdigung
"totes Pferd reiten"
2 Jahre
Zeitraum für radikale Änderungen
Einfach gelöst. it-agile
Moore's law
-> Prozessoren
Einfach gelöst. it-agile
"Shipping" CDs vs Online / SaaS
Consumer
Business
kunden-spez. Markt
einzelen Features neu
Big Bang vs. neue Features jeden Sprint
wo sind die alten Features?
Patches
alt
Birte
Soennecken eG
Soennecken-Platz
51491 Overath
www.soennecken.de
Soennecken
Herst.-Nr. 1140
Bestell.-Nr. 155 0211 02


Ankündigung
Einfach gelöst.
it-agile
Feedback Bubble
Einfach gelöst.
it-agile
Kommuni-kation
Einfach gelöst.
it-agile
Support
Einfach gelöst.
it-agile
Reaktion
Einfach gelöst.
it-agile
Release notes/ doc/ letter
Einfach gelöst.
it-agile
GUI mobile
Einfach gelöst.
it-agile
Verträge
Einfach gelöst.
it-agile
"Rollback-Requests"
Einfach gelöst.
it-agile
Termin?
neu
alt


Team für "alt" vs. "neu"
Einfach gelöst.
it-agile
Begeisterung für Neues
Einfach gelöst.
it-agile
"Legacy Software" (e.g. read only)
Einfach gelöst.
it-agile
Migration
Einfach gelöst.
it-agile
Konfigurations-migration
Einfach gelöst.
it-agile
Datenmigration
Einfach gelöst.
it-agile

Funktional geschriebene Software in Microservices

Vorteile "funktional":
- "Verlässlichkeit"
- Parallelisierung
- "if" ist teuer

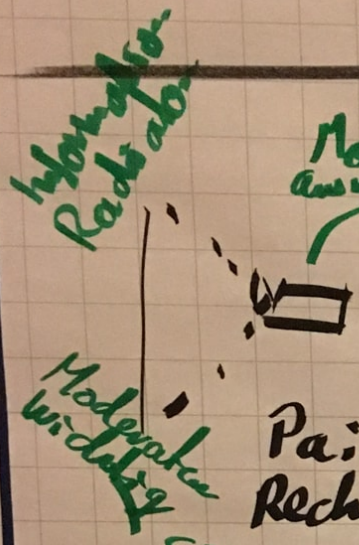
Quellen:
- Tiger-Team "SOA/ Micros."
- Michael Plöd DK DDD, Slideshare

Vorteile Microservices:
- Prototyping
- fachliche / User-orientierte Architektur
- Komplexitäts- / Lastverteilung

(Marco Heimeshoff) DDD mit F#

Nachteile / Gefahren:
- Gefahr "unnötiger" Komplexität
- „Vernünftige" Aufteilung der Domäne(n)
- Integrationstest sind evtl. sehr aufwändig
- passen Stories "immer" auf einen Service?

Mob-Programming



15' Rotation

Testen integrieren
- arbeiten dem Pair zu
⇒ Automatisierung
⇒ Wissensverbreitung

Fachseite dabei
⇒ schnelle Klärung

mündlich in Hörweite ⇒ Collective Code ownership

beschleunigtes Team-Building On-Boarding

welche Stories eignen sich?
- unsichere technische Lösung
- repetitive Dinge
- Bug Fixing
- Single-Piece-Pull

mit 50% anfangen?

explizites Check-out

am Ende kurze Retro

Wie verkaufe ich es meinem Kunden?
- Einzel-Buchführung
⇒ Langfristige Nutzen für Kunden

Wie überzeuge ich mein Team?
- Mob Friday
- 1 mal ausprobieren

TEAMS IN SELBSTORGA

VORRAUS.

- 1 PO, 1 PBL
  auch in Multiprojekt
- Team will selbstorga.
- Probleme, die ~~zur~~ durch selbstorga.
  gelöst werden sollen sind TEAM bekannt
- Ziel / GRAD d. selbstorga. ist abgestimmt
- aktueller Rolleninhaber wird zum Coach
- Team wird in Aufgaben gecoacht
- Befähigung
- Vertrauen in Team (muss auch in TEAM angekommen sein)
- Priorisierung einfordern
  - in sich selbst
  - in das TEAM
  - vom Mgmt. in das TEAM
- Zweck / Vision
- DELEGATION POKER -> Schrittweise in SELBSTORGA
- RETROS MIT TEAMLEITER UND/ODER Kunde

Ein Scrum Team - mehrere Projekte

Kein Team sondern Arbeitsgruppe?

Maintenance-Team mit Kanban Board für

Akquise, Support ... und andere Kleinigkeiten

Ziel vom Sprint => - Ruhe zum Arbeiten
- Ziel erreichen
- Störungen reduzieren

Redux - Patterns

state = {

user: ...

items

selectedProduct

:

}

sub reducer

- mapStateToProps

- functional Lib (z.B. ramda)

  hilft ungemein

Erweiterungen

- redux thunk ← einfach
- redux saga ← mächtiger, aber Komplizierter
- expect redux ← Interaktionen mit Store testen

Favourite Constraints

- Letzte Session:
  - Wiederhole dein Lieblingsconstraint
  - Choose your Constraint

- **Mute Session**
  - ↳ Vor Mittagspause :-)
  - ↳ Ping Pong / Evil-Coder Naiv-Coder

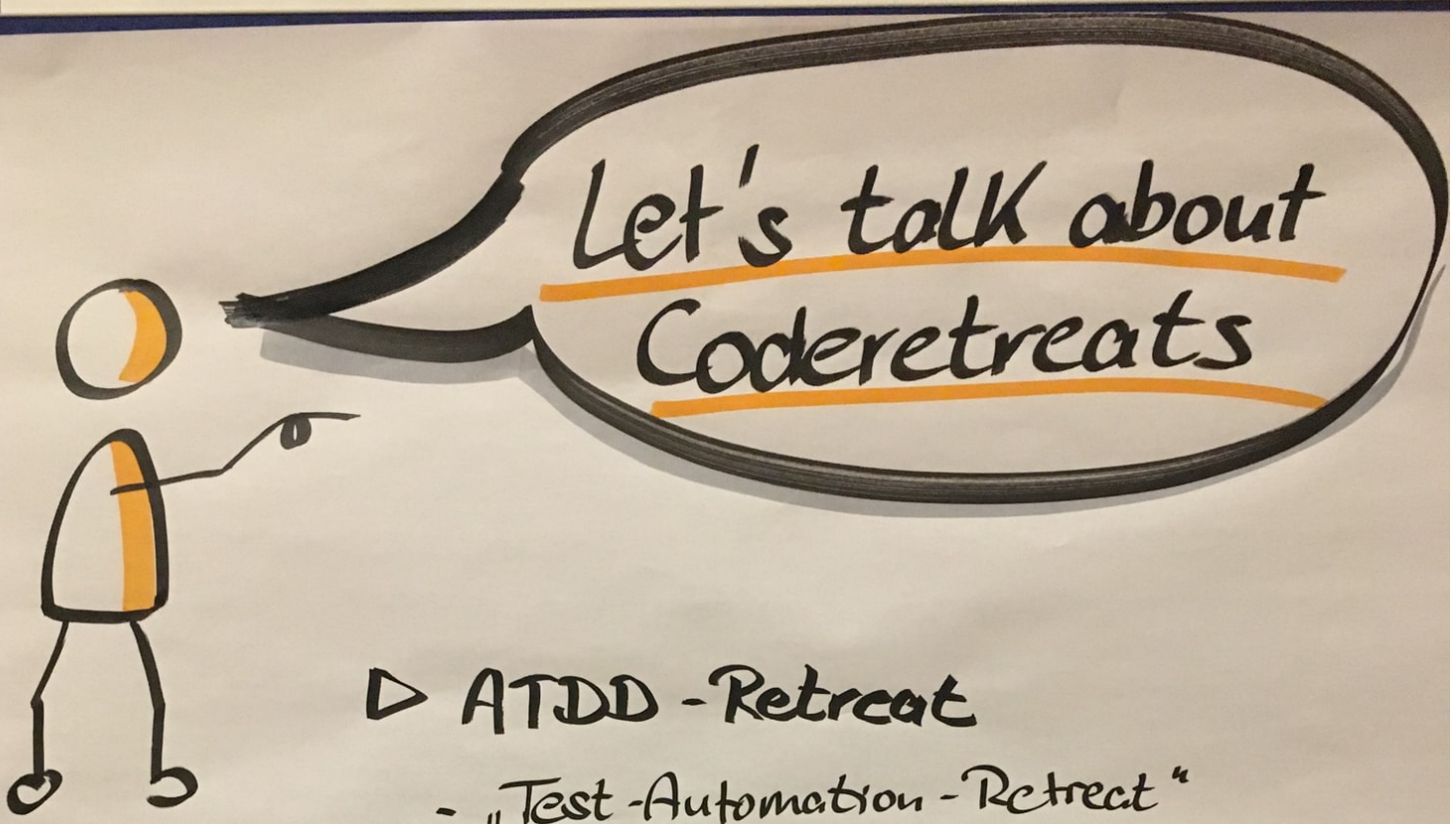
- **Baby Steps**
  - ↳ 2 min üblich
  - ↳ flexibilisieren anhand des Erfahrungslevels
    - ↳ Erfolgserlebnis
  - ↳ reset nach Zeitraum automatisieren :-)

- Object Calesthenics

- Mob-Programming + unbekannte Programmiersprache

Generell Wichtig:
- Leute nicht mit zuviele Constraints überfordern

ANKEN



▷ ATDD-Retreat

- „Test-Automation-Retreat"
- gut vorbereitet

Erfolgsfaktoren
- gute Vorbereitung des Beispiels
- Mischung von Erfahrenen / Neulinge

Beispiel ~~objekt~~ projekt
↳ Existierende Codebasis?
↳ Techniken mit PO-Challenge pairen
↳ Testautomatisierungs-retreat
↳ Neue Techniken

FizzBuzz

45 m.

gut!

- IntelliJ + Refactoring − Test Execution
- Publikum aktivieren
- Parameterized Tests

einfach

- Integer.~~valueOf~~ toString > String.valueOf
- Guards vs. Geschäftslogik

```
if (... // guard
}
```

- Duplikation vs. Einfachheit

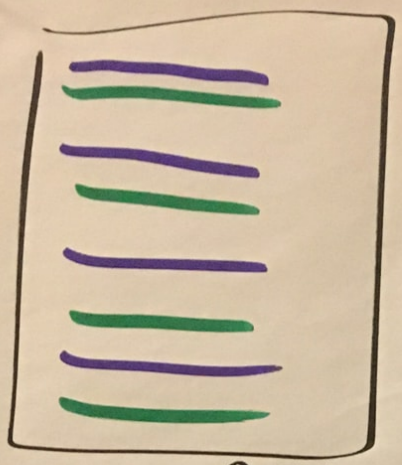
Won

- Isomorphie Business Logik ≅ Code

3 Fizz → 3 Buzz

Full Stack Developer

* neue Plattform
  vs. bestehende Codebasis
* cross-functional-team:
  - Vorsicht bei Aufgabenverteilung
    (techn. Inseln)



LeSS

Huge
→ 8 Teams

Requirements Area
Product Owner

4-8 Teams

Basic LeSS

4-8 Teams

Basic LeSS

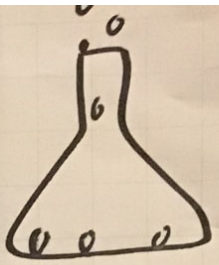
Communities — Community Lead

Architecture & Design

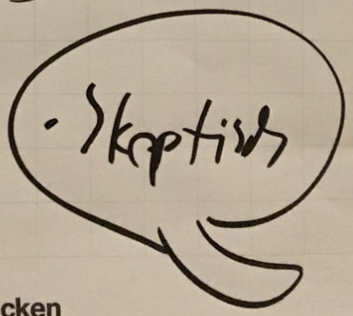
Scrum Master

1 Team auf rotierender Basis arbeitet an Verbesserungen

FizzBuzz II



- deskriptive Darstellung
  - etwas kryptisch
- Pattern Matching ohne Typisierung
- getrennt Datenstrukturen & Funktionen
- neue Syntax
- Pattern Matching ist cool
- Ruby kills {} ()
- FizzBuzz!



Informations-häppchen

Stand Up

Neue Tasks

Infos als relevant markieren

Mal schnell schauen

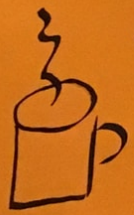
Build Status

Kommentare (@-mention)

Suche

Google mini

Morning Coffee



offene Pull-Requests

Pull oder Push ?

Chat, Mail, Jira, Microblog

langlebige vs. kurzlebige Informationen

Personal Assistent

Nicht Einzelevents sondern Statuswechsel

Filter

Schlaue Filter

Rauschen

Soennecken eG
Soennecken-Platz
51491 Overath
www.soennecken.de

Herst.-Nr. 1140
Bestell.-Nr. 155 0211 02