



Ist das Kunst oder kann das weg?

Software-Architektur im agilen Projektumfeld



Individuelle Anwendungsentwicklung - Java Enterprise, Web, Mobile

- seit 2005 ♦ in Köln ♦ für alle Branchen ♦ **Goldschmiede**  anderScore
- nach Aufwand & zum Festpreis
- ✓ Digitalisierung / Prozesse / Integration
- ✓ Migration
- ✓ Neuentwicklung
- ✓ Notfall / kritische Situation
- ➔ pragmatisch, zielgerichtet, zuverlässig

Kompletter SW Life Cycle

- Projektmanagement / agile Methodik
- Anforderungsanalyse
- Architektur & SW-Design
- Implementierung & Testautomation
- Studien & Seminare



... und für Sie? Sprechen Sie uns an!

Patrick Loka (M.Sc. Informatik)

- Senior Software Engineer
 - Projekte, Consulting, Trainings
 - Vorträge, Workshops, Artikel
- Technisch
 - (Java) (Web) Development
 - Migration & Integration
- Rollen
 - Software-Architekt (iSAQB: CPSA Advanced Level)
 - Senior Developer
 - Anforderungsanalyst
 - Agile Coach (scrum.org: PSM I)
- Sonstiges
 - Theoretische Informatik
 - Mobile



Architektur

Was war das noch gleich?



Architektur

Was war das noch gleich?



Architektur

Was war das noch gleich?

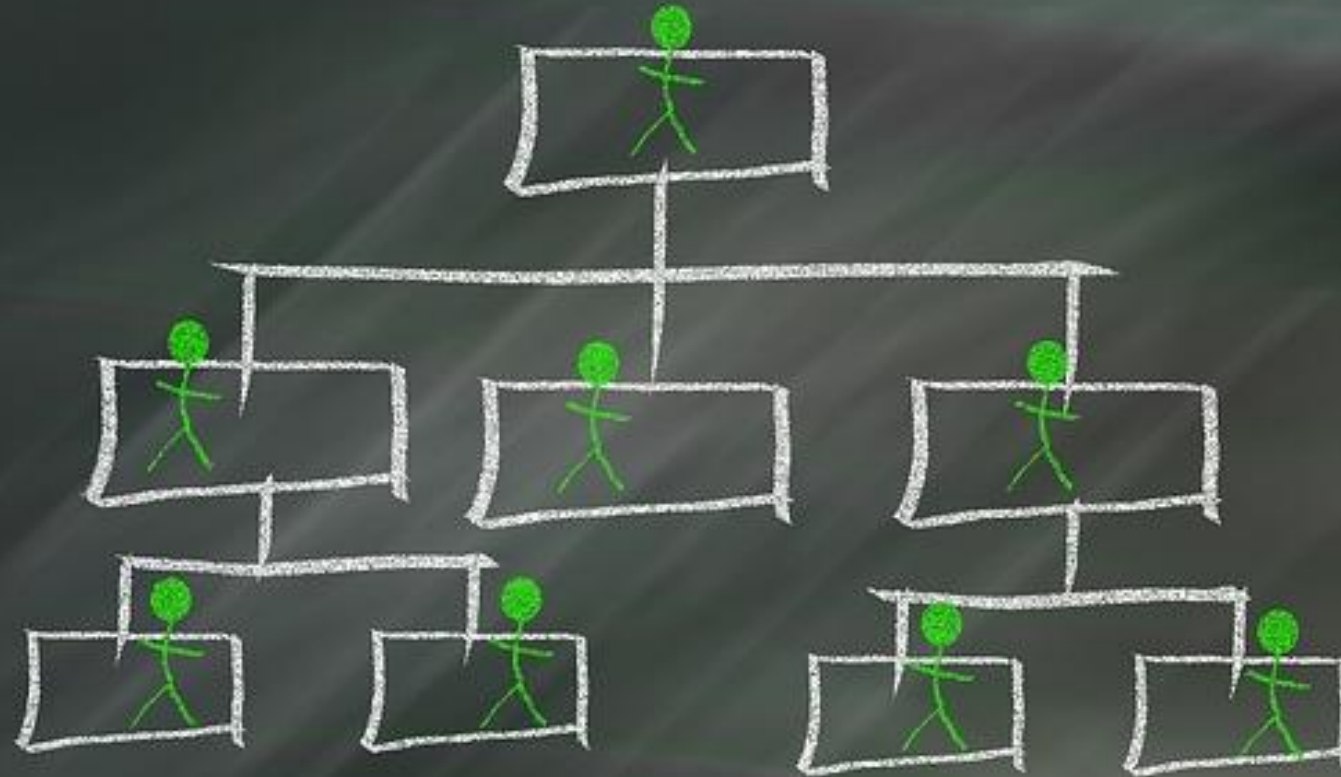
**Must
have**

Technologie Driven Architecture



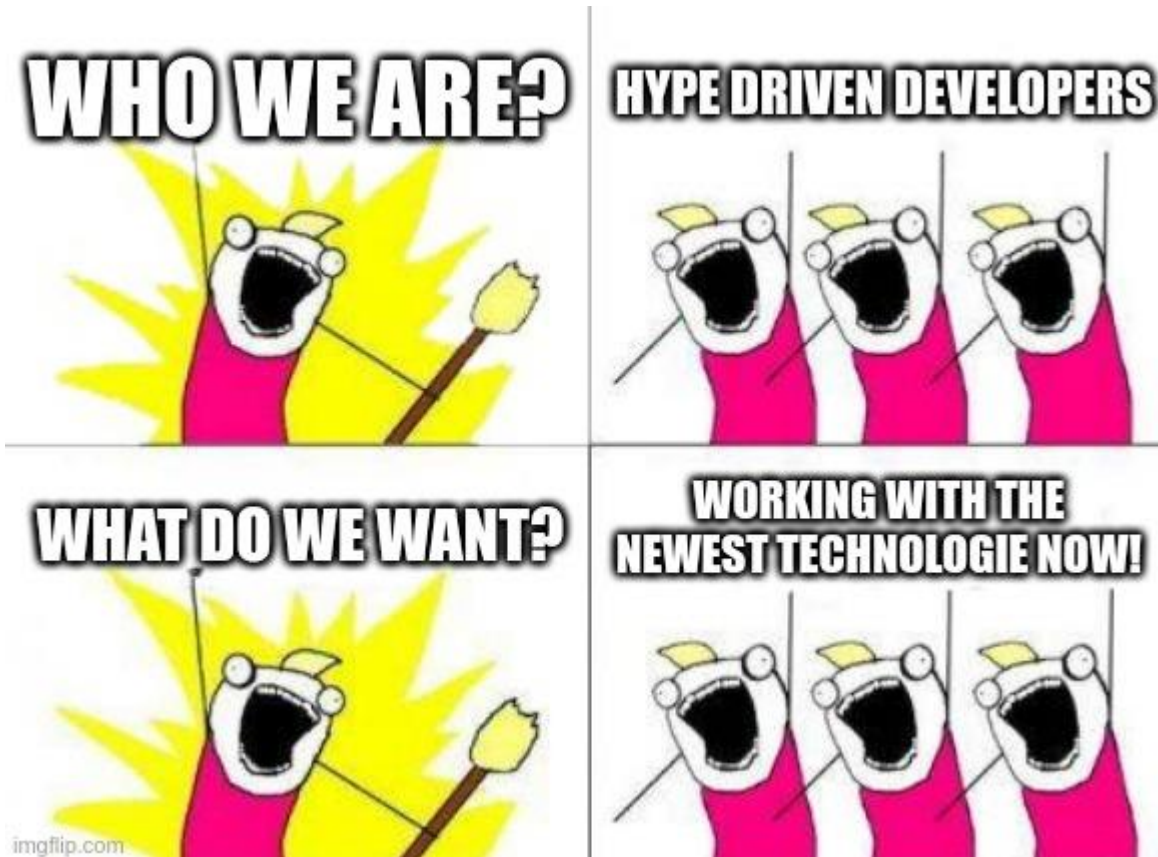
Architektur

Was war das noch gleich?



Conway's Architecture

Hype Driven Architecture



1. Problem und Lösung
2. Vorstellung, Buzz- und Keywords
3. Hype beginnt
4. Enttäuschung
5. Realisierung

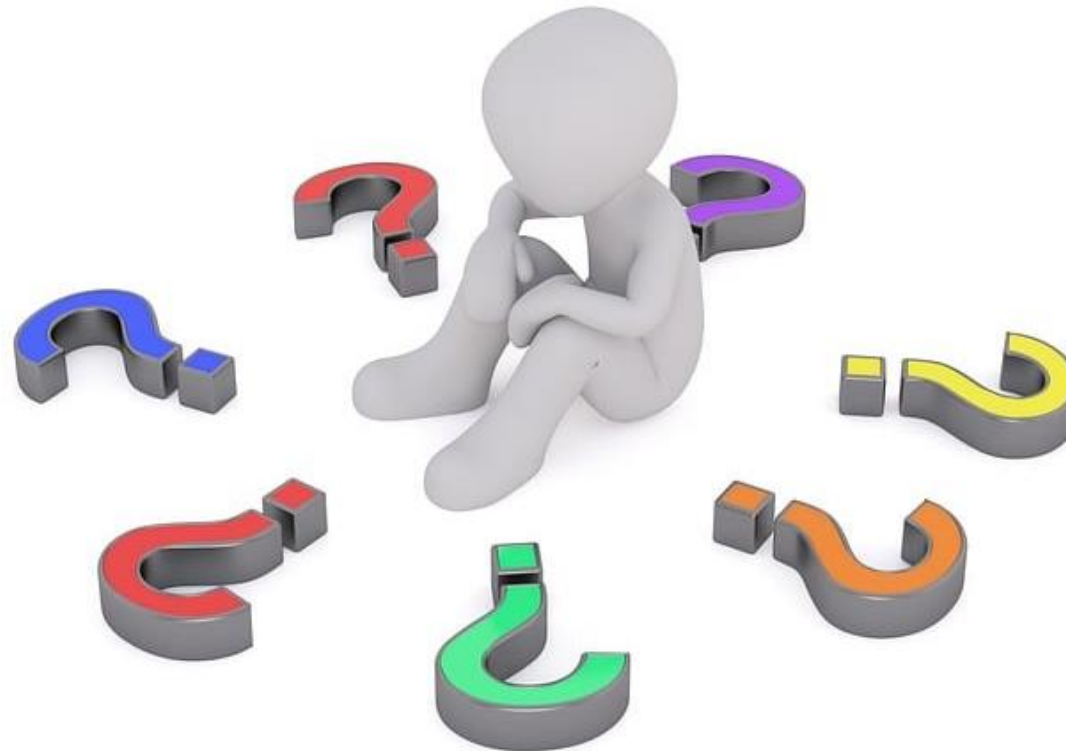
Architektur

Was war das noch gleich?

It worked for Google
Architecture



Warum machen wir das eigentlich?





- Fachlichkeiten
- Mehrwert
- Geschäftsziele

Architektur

Rahmenbedingungen einhalten



- Projektmanagement
- Technisch
- Juristisch, ...



- Subjektivität
- Qualitätsmerkmale
- Quantifizierung

ISO 25010 – Qualitätsmerkmale

(update coming soon...)

Funktionalität

- Vollständigkeit
- Korrektheit
- Angemessenheit

Effizienz

- Zeitverhalten
- Ressourcenverbrauch
- Kapazität

Kompatibilität

- Ko-Existenz
- Interoperabilität

Benutzbarkeit

- Erlernbarkeit
- Bedienbarkeit
- Schutz vor Fehlern des Benutzers
- Barrierefreiheit

Zuverlässigkeit

- Reife
- Verfügbarkeit
- Fehlertoleranz
- Wiederherstellbarkeit

Sicherheit

- Vertraulichkeit
- Integrität
- Nachweisbarkeit
- Ordnungsmäßigkeit
- Authentizität

Wartbarkeit

- Modularität
- Wiederverwendbarkeit
- Analysierbarkeit
- Änderbarkeit
- Testbarkeit

Portabilität

- Anpassbarkeit
- Installierbarkeit
- Austauschbarkeit

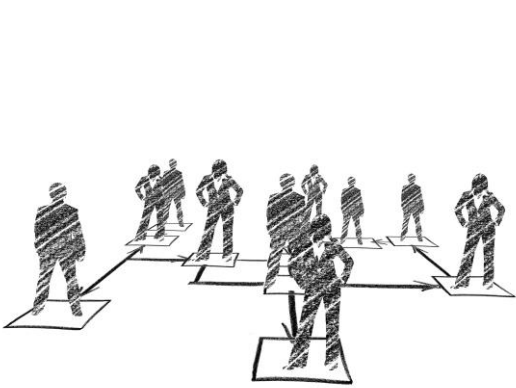
Qualitätsszenarien

(Beispiel Änderbarkeit)

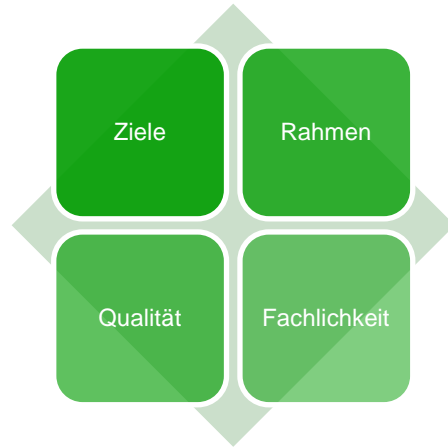
- **Ereignis:** Vorstandsbesetzung ändert sich – Vorstandszeile in Dokumenten muss angepasst werden
- **Quelle:** Juristische Abteilung
- **Kontext:** Die Änderung wird implementiert
- **Artefakt:** Dokumentenerzeugung
- **Antwort:** Es werden zu keinem Zeitpunkt zwei Dokumente mit unterschiedlichen Vorstandszeilen erzeugt, die Änderung soll keine Seiteneffekte haben
- **Maß:** Änderung soll innerhalb von 1 Tag durchgeführt, getestet und in Produktion sein

Architektur

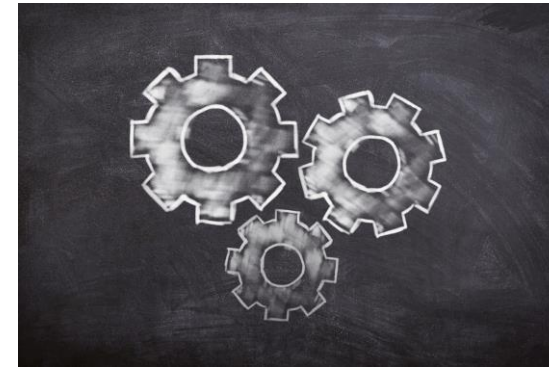
Anforderungs Driven Architecture



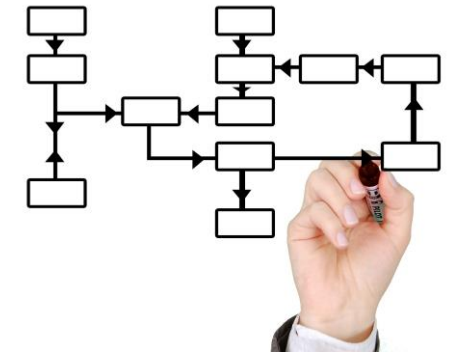
Stakeholder



Einflussfaktoren



Architekturarbeit



Lösungsentwurf



Agilität

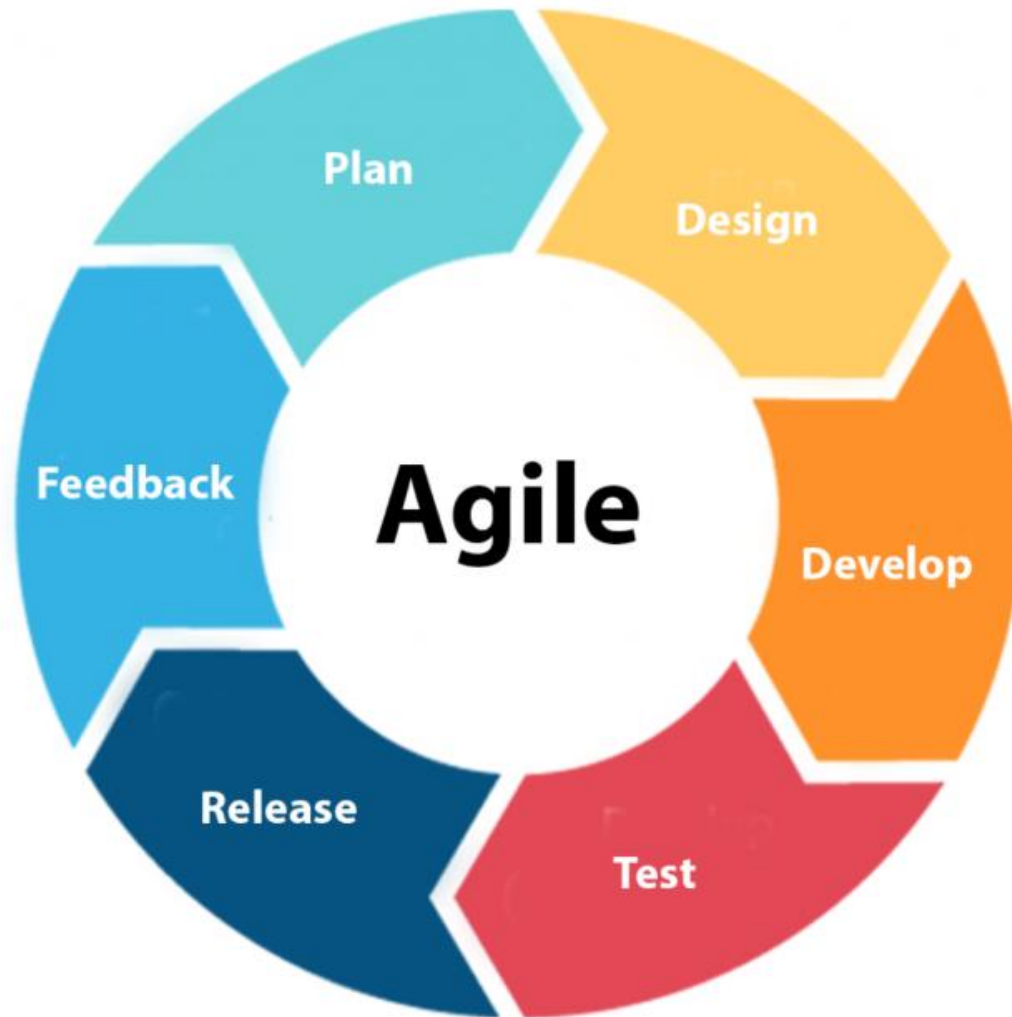
Wofür macht man das nochmal?



- Time to Market?
- Ungewissheiten?
- Kontinuierliches Feedback?
- Nachbesserungen?
- Kontinuierliche Weiterentwicklung?

Agilität

Wofür macht man das nochmal?



- Dynamische Einflussfaktoren?
- Arbeit in Zyklen?
- Arbeit in diversen Teams?
- Wer macht Architekturarbeit?

Scrum

Ein Team die Welt zu retten!



- Product Owner
- Scrum Master
- Team (Self Contained)

Scrum

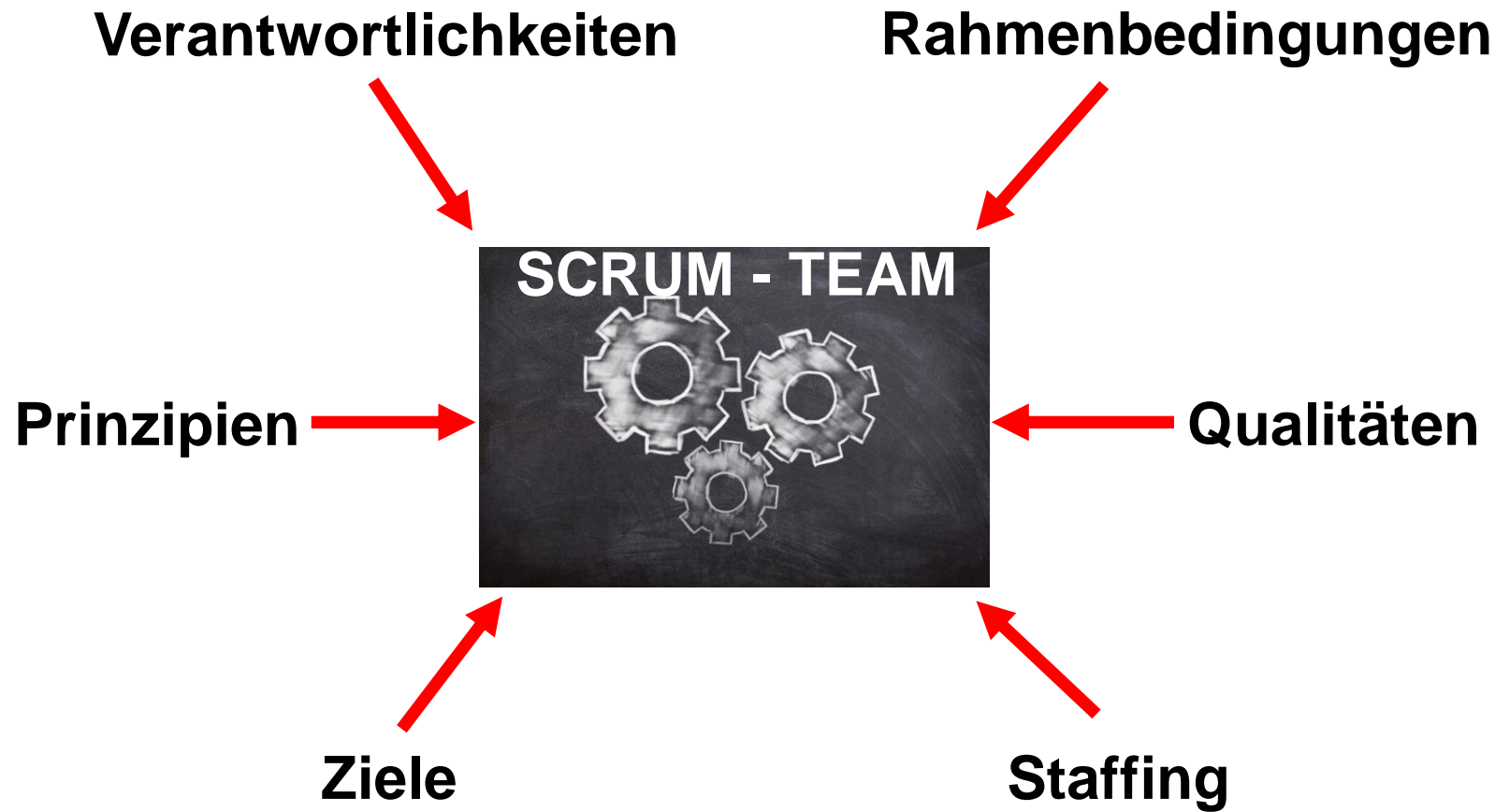
Ein Team die Welt zu retten!



- Blackbox
- Scope
- Entscheidungen

Scrum

Ein Team die Welt zu retten!



- Aufgabe (Self Contained)
- Kontext
- Beschränkung
- Befähigung
- Lenkungswirkung
- Optimierungsvektor

Scrum

Ein Team die Welt zu retten!

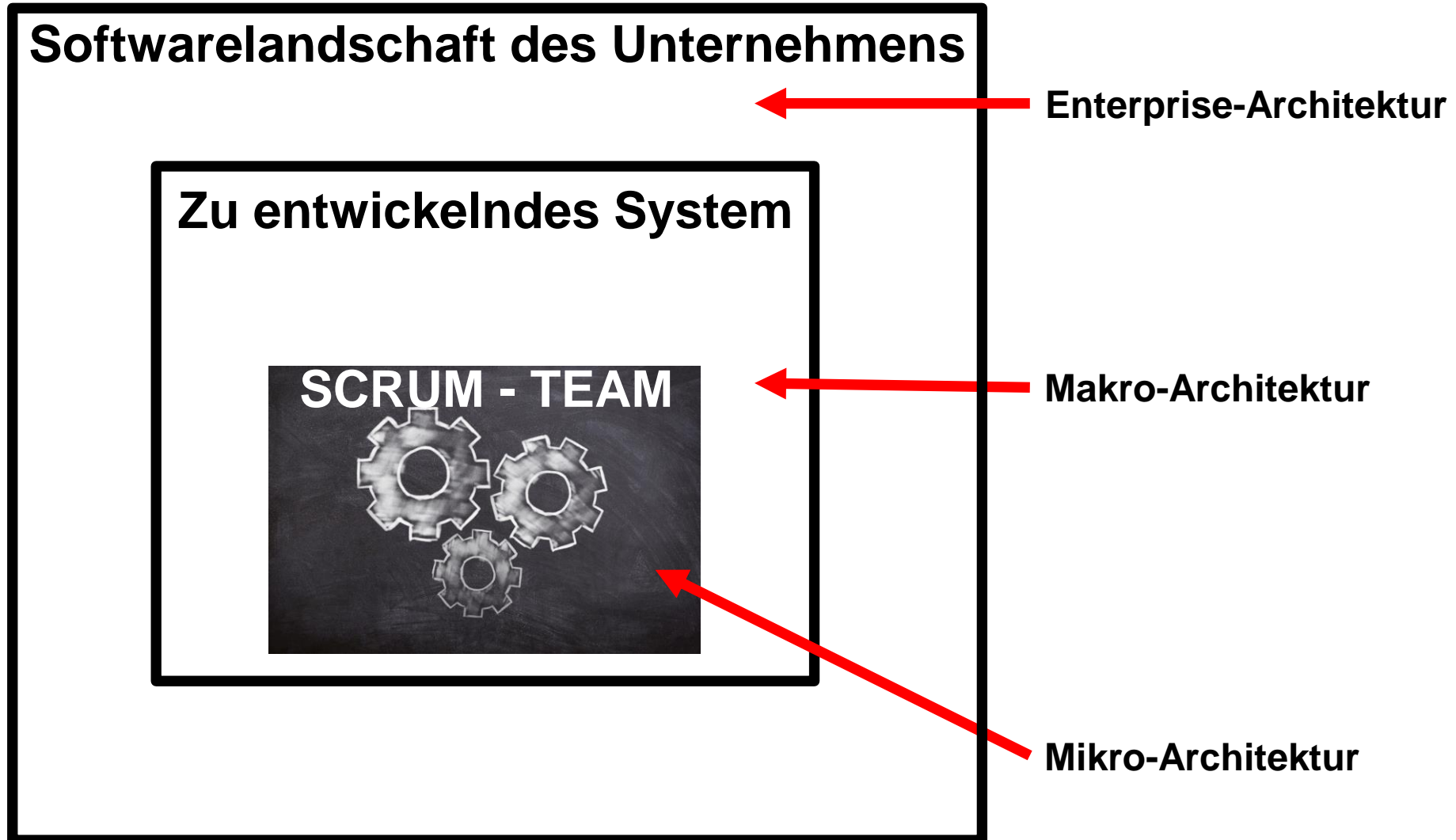


- Diversität
- Kompetenz vs. Anforderung
- Agiler Zyklus abbilden
- Scope?

Wo ist der Architekt?



- Durchs Team ersetzt?
- Teil des Teams?
- Berater des Teams?
- CoP / Veedel?
- Stakeholder?
- ...



Architekturarbeit

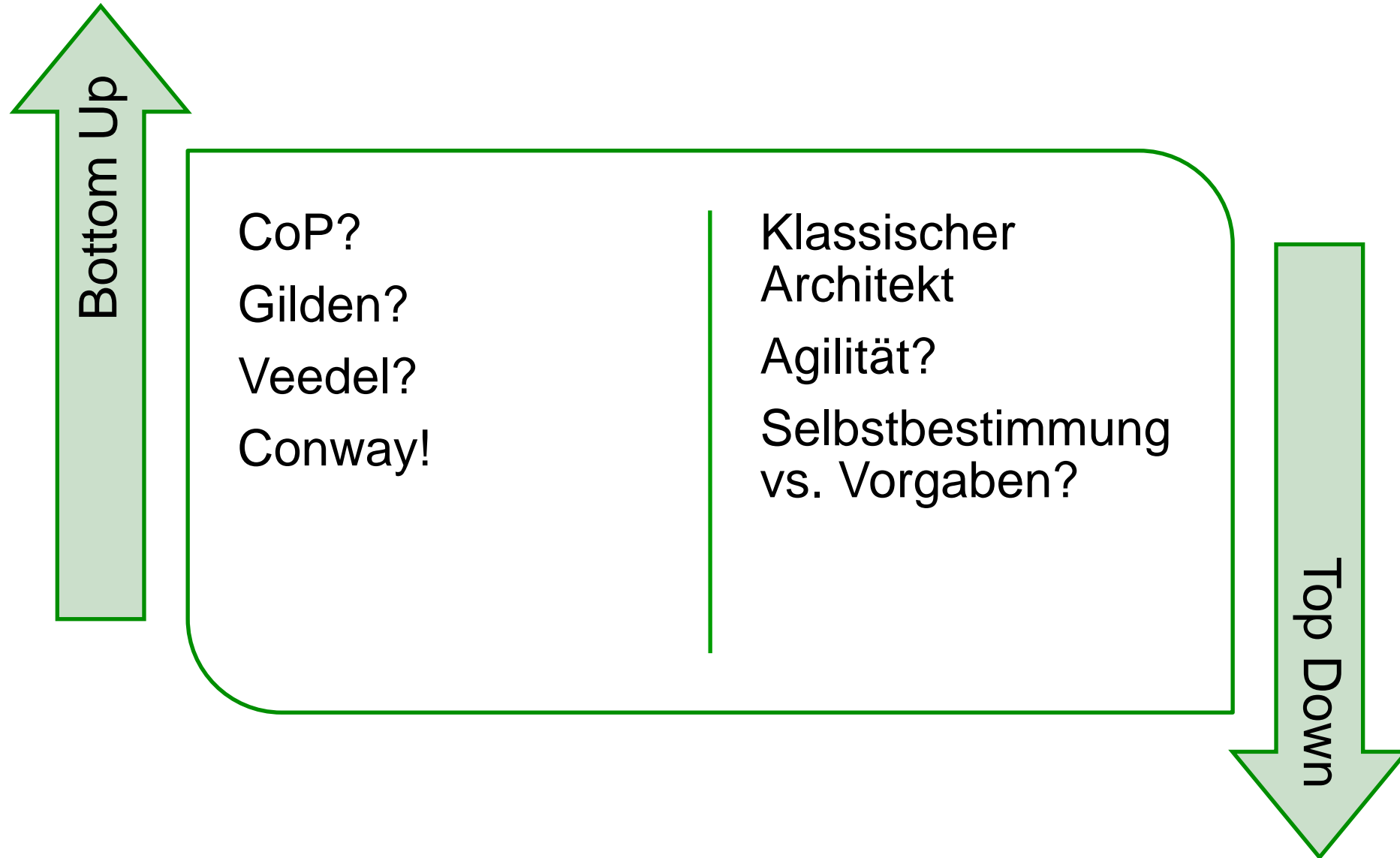
Warum Makro-Architektur?



- Mehrwert durch System herstellen
- Rahmenbedingungen des Systems einhalten
- Qualität des Systems sicherstellen

Architekturarbeit

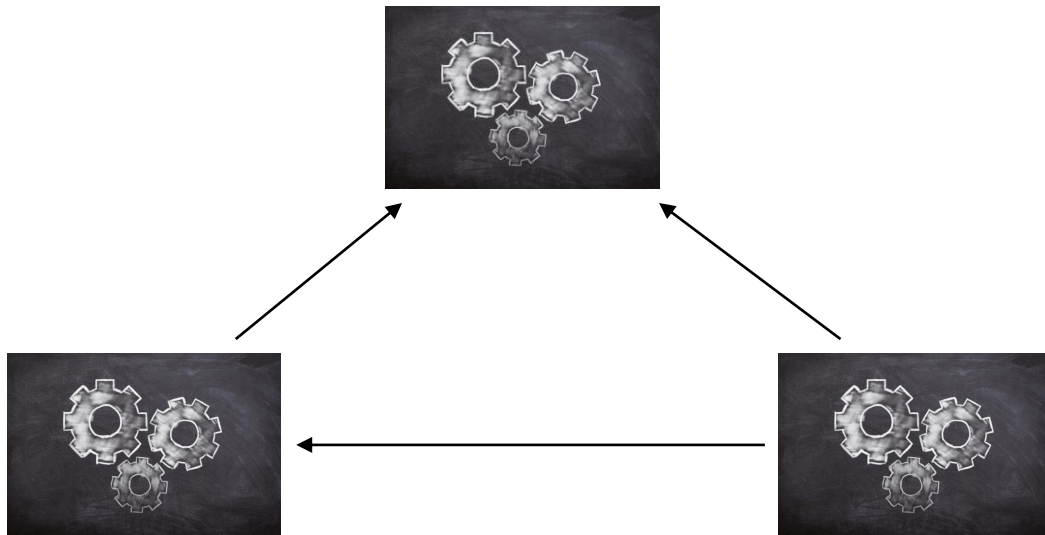
Wer macht Makro-Architektur?



Architekturarbeit

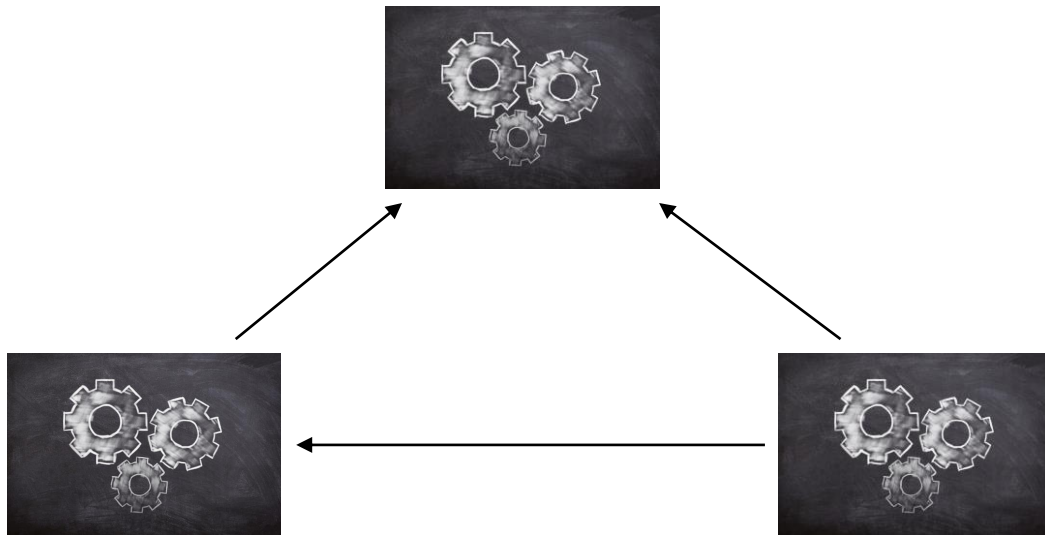
Was darf Makro-Architektur?

System



- Querschnittskomponenten festlegen
(*Was ist Makro / Mikro?*)
- (Self Contained) Blackboxen definieren
(*Ziele, Qualitäten, Rahmen, Verantwortung, Prinzipien, Staffing*)
- Anforderungen an Blackbox – keine Vorgaben!
(*User Stories, Qualitätsszenarien, Evil User Stories, ...*)
- Beziehungen zwischen Blackboxen festlegen
- Blackboxen Testen
- Conways Law ausnutzen!
- => Entwurf eines Komponentendiagramms

System



- Scrum Team für System mit System PO
- Diverses Staffing
- Keine Hierarchie – Anderer Scope!
- Integriert in Agilen Zyklus
- Anforderungen ans System
- Anforderungen der Blackboxes

Architekturarbeit

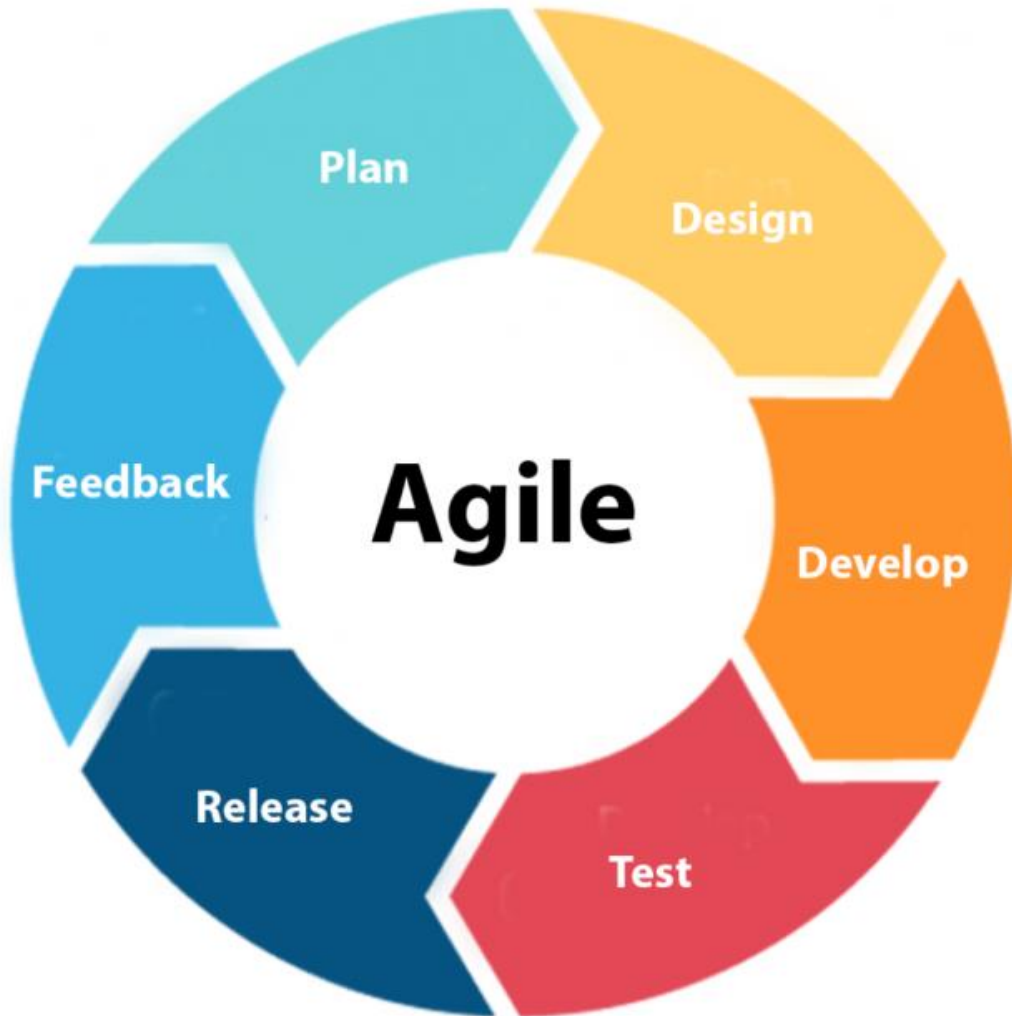
Ist der Architekt also überflüssig?



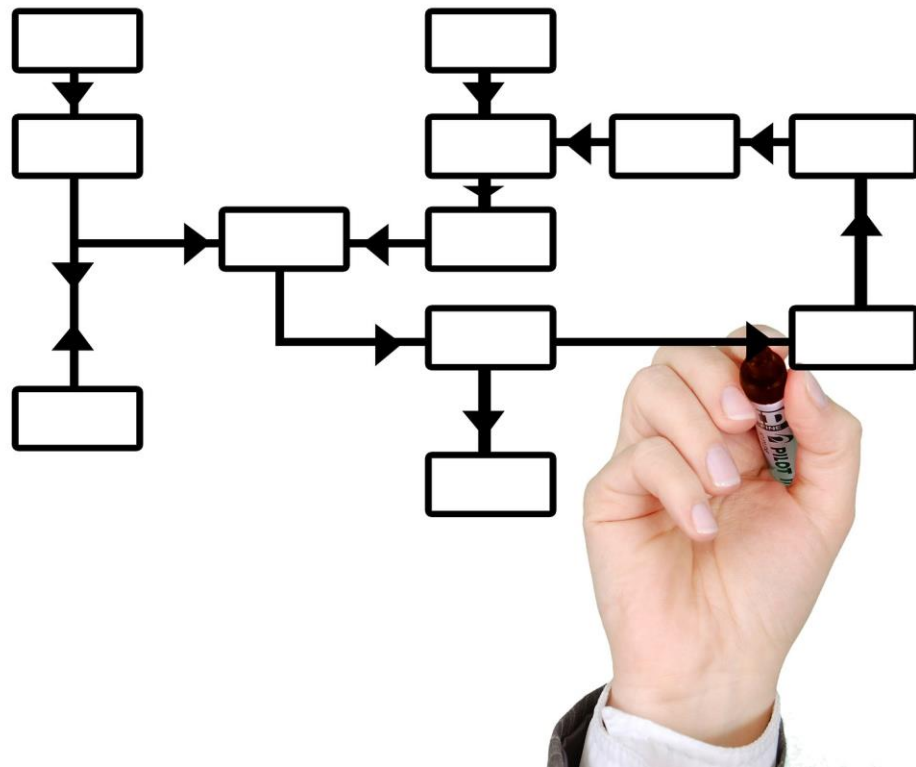
- Was ist Aufgabe des Teams?
- Wie hoch sind Qualitätsansprüche?
- Kann Architekturarbeit vom Team abgedeckt werden?
- Wie viel Architektur gibt der Kontext vor?

Agilität

Das Problem mit dem Zyklus

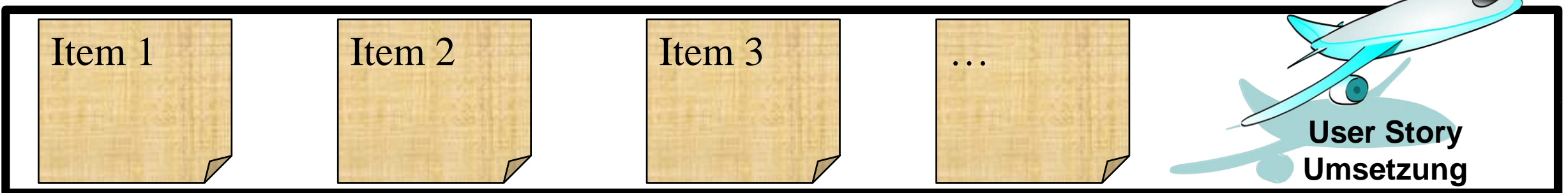


- Dynamische Einflussfaktoren
- Sprint Scope
- Mehrwert Fokussierung



- Agile Entwicklung von Architektur
- Grober Architekturentwurf (SOLL!)
- Deckt bekannte Einflussfaktoren ab
- Gibt Kontext und Richtung für tägliche Arbeit
- Entwickelt sich mit Anforderungsänderungen und –Detailierung
- Niemals fertig, Ständig hinterfragen!
- Vgl. Dokumentation (IST)

- Was für Architektur Komponenten benötigt eine User Story?
- Architektur-Anforderungen sichtbar machen
- Wie lange könnte es bis zur Umsetzung der User Story dauern?
- Architektur Items anlegen, Schnittmengen zu anderen User Stories identifizieren, Zielarchitektur anpassen
- User Story realistisch (ein-)planen





- PO, Team, Stakeholder...
- Teil von User Stories
- Verbesserung von Qualität
- Änderung der Rahmenbedingungen
- Durchlaufen den ganzen Zyklus (Test)
- Qualitätsszenarien
- Architecture Decision Records

Architecture Decision Records

(Beispiel Pdf Dokumente Persistenz)

- **Name:** Persistenz von Pdf Dokumenten
- **Status, Datum:** Entschieden, 03.02.2023
- **Kontext (Problem, Motivation):** Im Rahmen des Prozesses soll der Kaufvertrag mit dem Kunden Nachweissicher gespeichert werden
- **Möglichkeiten (Pros & Cons):** Dokumentstore, RelDB
- **Entscheidung (Begründet):** Dokumentstore, weil...
- **Konsequenzen:** Dokumentstore muss Aufgesetzt werden, Transaktionssicherheit über 2 Systeme muss gewährleistet werden, ...



Vielen Dank!

Und welche Erfahrungen haben Sie gemacht...?

