

Continuous Architecture für agile Teams

Objektforum, 28.03.2017

Christian Bader – Agile Coach/Scrum Master

Uwe Kranz – Agile Software Engineer

Agile Architektur – ein Widerspruch?



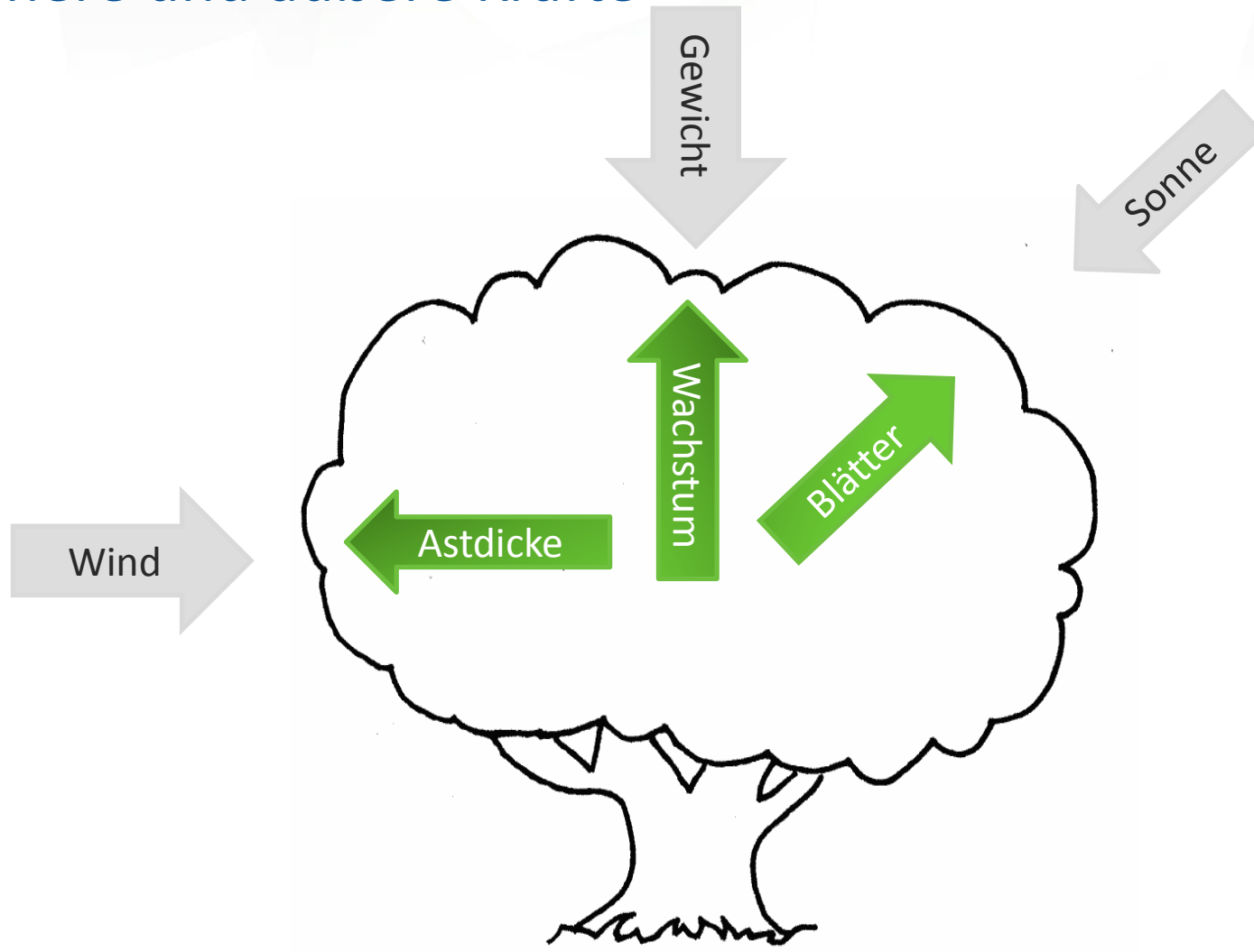
Architekten, um 1910
Foto: Gallica, France | 01. Februar 1919



Modellierte Baumarchitektur – Wind



Innere und äußere Kräfte



Nährstoffe, Wasser, Bodenbeschaffenheit



Zyklen

- Jahreszeiten
- Trockenheit/Feuchtigkeit
- Schaden/Heilung



Zyklen

- Tag/Nacht
- Sonne/Schatten
- Wind/Windstille



Versorgung

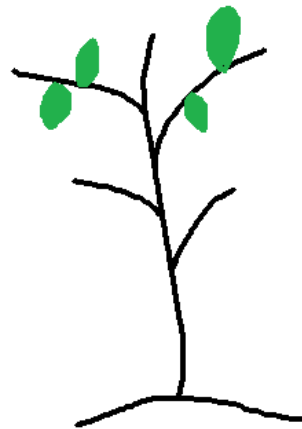
- Sonne
- Sauerstoff und Kohlendioxid
- Wasser
- Nährstoffe
- Bodenbeschaffenheit



Wertgetriebene Architektur

- Kundenwert wird inkrementell geliefert

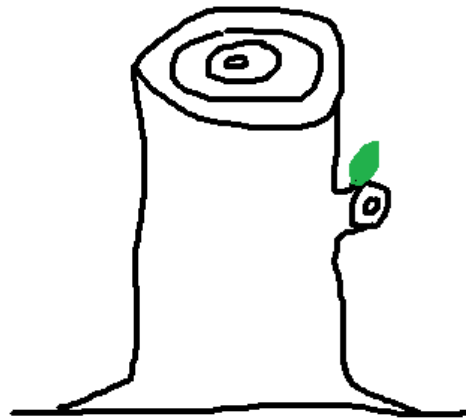
WILL LIVE



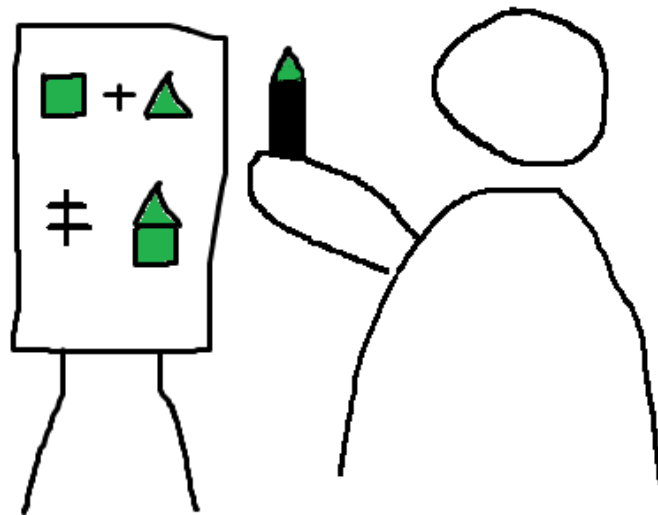
Klassische Architektur

- Klassische Architektur ist plangetrieben

POOR

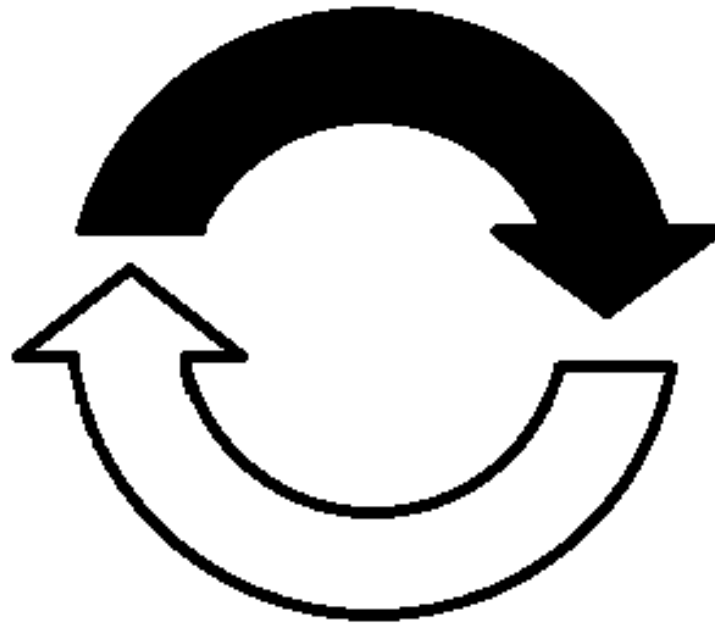


Case Study - Performance



Was ist das Problem?

No info, can't plan

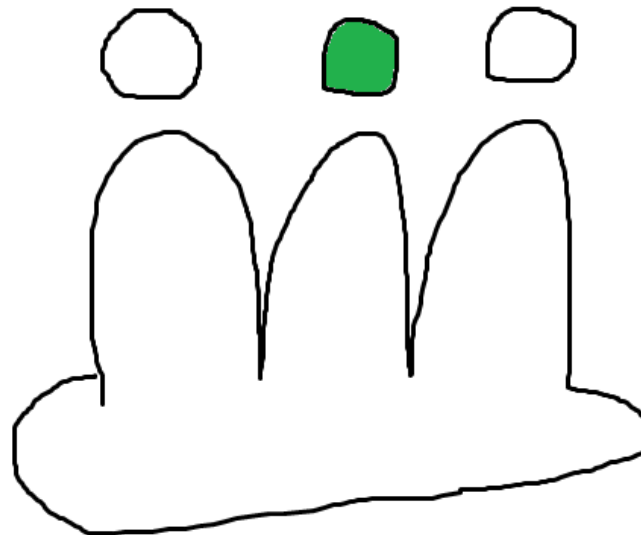


No plan, can't start



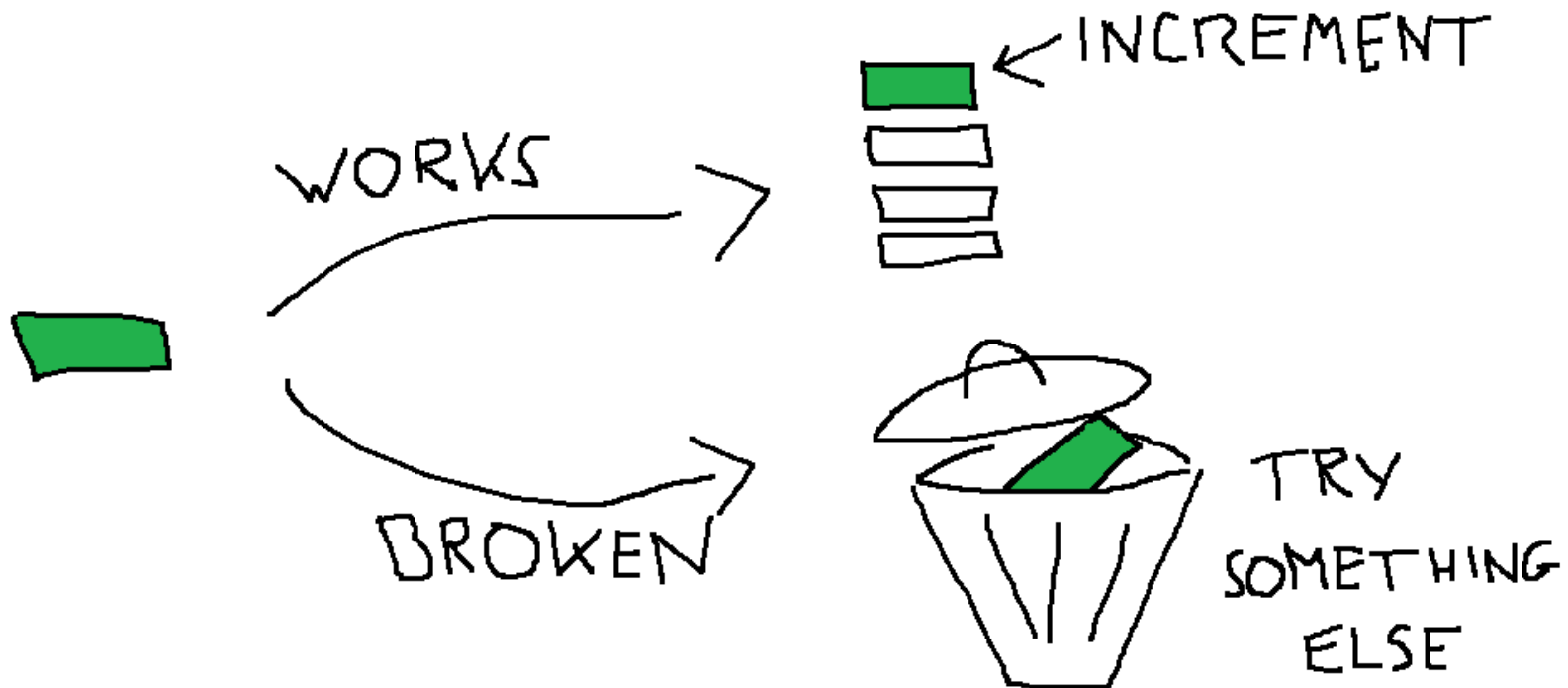
Kommunikation

- Agile Architektur entsteht im Team



Produktfokus

- Agile Architektur wird im Produkt inkrementell bestätigt oder verworfen



Transparenz über äußere Kräfte

- Product Backlog, User Stories und Akzeptanzkriterien
- A/B-Testing
- Metriken zur Verwendung des Systems
- Betriebswirtschaftliche Metriken
- Gespräche mit Fachexperten
- User Feedback



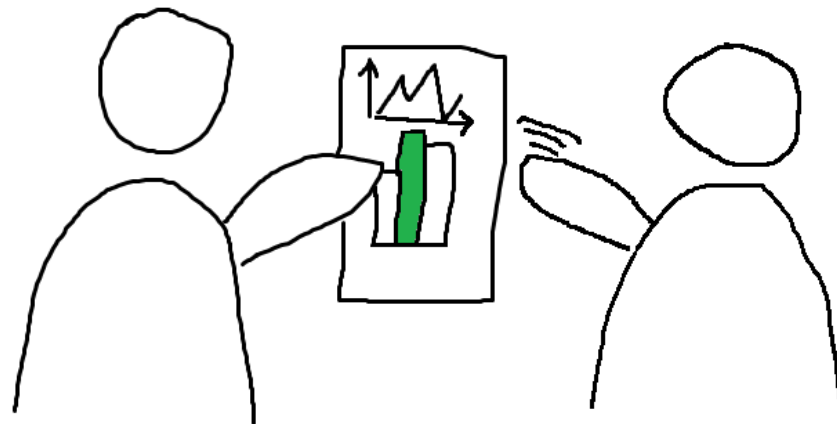
Transparenz über innere Kräfte

- Pair Programming
- TDD, Testing
- Collective Code Ownership



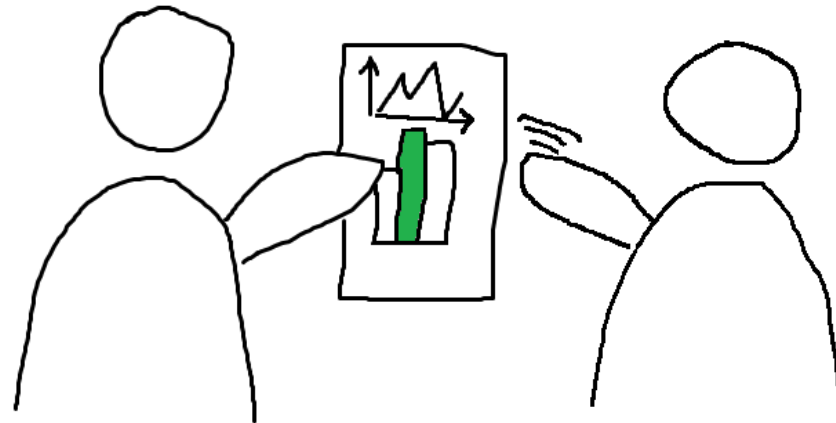
Metriken

- Kontinuierliche Planungshilfe
- Kommunikationsmittel, Indikator
- Automatisierung, IDE-Integration



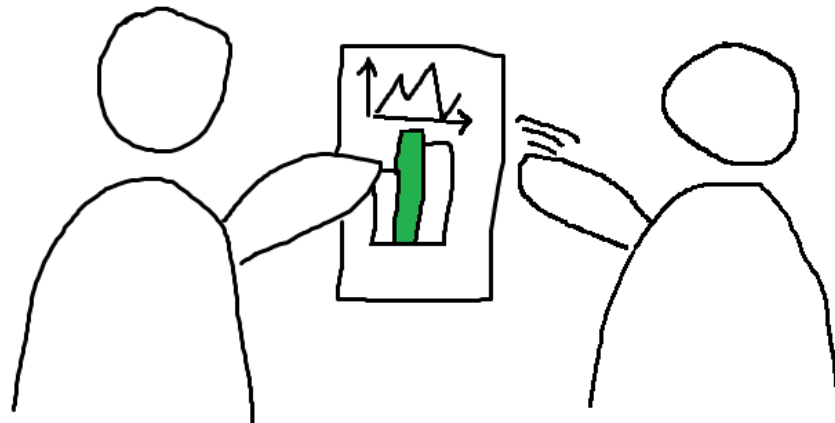
Metriken

- Klassengröße
- Methodenlänge
- Testabdeckung

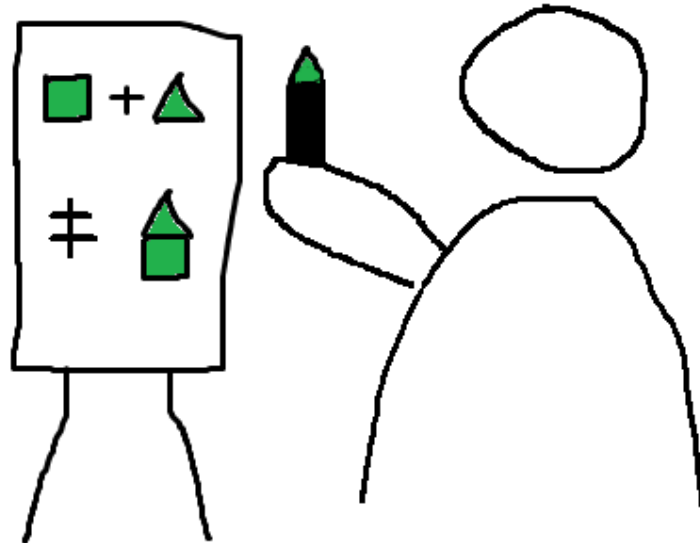


Metriken

- Paketzykel
- Zyklomatische Komplexität
- Kopplung



Case Study - Metrics



Feedback-Schleifen

- Pair Programming
- Compile
- Build



Feedback-Schleifen

- Daily Scrum
- Sprint
- Release
- Produktion

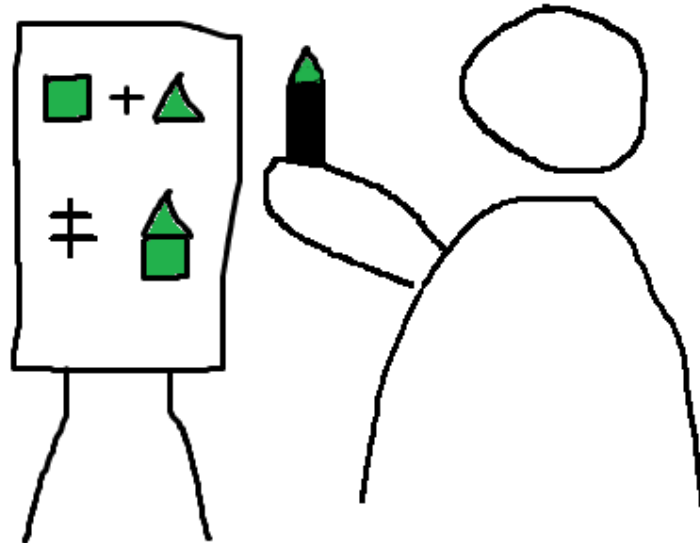


Beschleunigung des Feedbacks - DevOps

- Continuous Integration
- Continuous Delivery
- Continuous Testing & Monitoring



Case Study - Feedback



Unsere Erfahrungen

- Früh (nach 2-3 Sprints) und kontinuierlich Software an den Endkunden/Benutzer liefern, um Feedback zu erhalten
- Agile Prinzipien beachten (nicht nur Scrum ;-)
- Enge Zusammenarbeit & Kommunikation im Team
- Know-How des Teams und eingesetzte Software müssen passen (falls nicht temporär Experten ins Team)
- Architekt/Senior Software Engineer entwickelt im Team mit (Fähigkeiten statt Rolle) und hat Erfahrung in der agilen Softwareentwicklung
- Mit wenigen Experten (in der Domäne, Methode und Tools/Software) praktisch anfangen und langsam weitere Teammitglieder aufnehmen



Unsere Erfahrungen

- Architektur kontinuierlich (immer wieder!) auf verschiedenen Granularitätsebenen stattfinden lassen.
- Software in alle Richtungen wachsen lassen (User, Daten, Anzahl Umgebungen, Funktionalitäten, Performance, Tools, Komponenten, Testpyramide, etc.)
- Optionen erhalten = späte und möglichst kleinschrittige Entscheidungen
- Das Rad nicht neu erfinden: Doku, Paper, Konferenzen, Social Media, etc. lesen und Experten fragen.
- Prinzipien & Pattern nutzen (z.B. SOLID, 12Factor, etc.)
- Kontinuierliches Refactoring (muss das Team beherrschen oder lernen)



Unsere Erfahrungen

- Einfache Metriken (Software, Prozess und Business) von Anfang integrieren
- Fraktal arbeiten (wann schneide ich die Architektur wo?) -> sich daran gewöhnen und üben!
- Test-Driven entwickeln
- Build- und Delivery-Kette gehört zur Architektur der Anwendung
- In Refinement, Sprint Planning und beim Auftauchen von Hindernissen im Sprint Architekturmethoden zur Visualisierung nutzen
- User Stories schneiden (immer vertikal!)



Transparenz & Feedback treiben agile Architektur

Gute Architektur entsteht mit Transparenz über:

- Äußere Bedingungen & Anforderungen
- Innerer Zustand des Systems

... mit Hilfe von Scrum, XP und DevOps zur Schaffung von Feedback-Schleifen



Start small & grow it!



Fragen & Diskussion in Kleingruppen

- Ziel:
 - Erfahrungsaustausch durch moderierte Diskussion in Gruppen.
 - Zuhören & Mitreden
 - Ideen vermitteln bzw. kennen lernen



Fragen & Diskussion in Kleingruppen

- Ablauf:
 - Themen bekannt geben
 - Aufteilung in 4 Gruppen
 - Pro Gruppe einen Moderator festlegen
 - 20 Minuten Zeit
 - Im Anschluss Treffen im Plenum und Abschlussrunde (Flipchartständer bitte mitbringen)



①

Rolle des
Architekten in
der agilen Welt
①

- * „heirlicher“ Architekt
- * Architekt als Coach des Teams
- * Architekt als „technischer PO“
- * Regelmäßige Meetings zur Diskussion von Architektur-Entscheidungen
- * Entwickler stellen techn. Stories ein

②

Agile Architektur
und
Großprojekte
②

- Bestands arch. ### |||
- Eigenst. Arch.-Team + Agile Teams + kompetenter !!
- Quality Gates |||
- Kollaboration zw. Teams + Effektivität/
effizient // ##
- ↳ Kommunikation zw. Architekten
↳ + Dev + Stakeholder
- Team-Schnitt ### |||



Experts in agile software engineering

