

Antipattern bei agilen Transitionen

Track Agile
Von 10:45 bis 11:30 Uhr
21. Februar Conference Day
Frankfurter Entwicklertag 2018 #etffm18

Frank Gollas (@frankgollas)
Stefan Görgner (@klaaaatuuuu)
Patrick Sauerwein (@virtual_patrick)
Marcus Schreiber (@MarcusSchreibe4)

Kontext der Erfahrungen

Unsere gemeinsame Erfahrung

“Agile Transition” im mittelständischen Softwareunternehmen

- Vier Jahre agile Transition (Zeitraum **2013-2017**)
- Vier Produkte (Cloud, Mobile; **On Premise**)
- Vier Entwicklungsstandorte (Kiew, Sydney, Karlsruhe, **Frankfurt**)
- Insgesamt 12+ Teams (**4-6 Teams**)
- Durchschnittliche Teamgrößen ca. 5-9

Warum “agile Transition”?

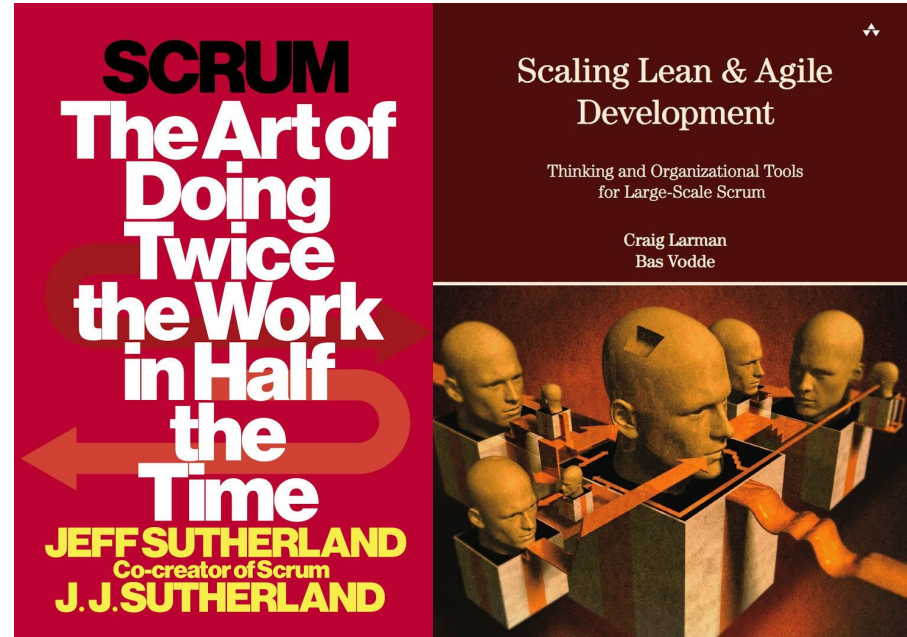
Herausforderungen

- Time to market (max. 0 bis 1 Release pro Jahr)
- Wettbewerb schneller im Markt
- Hohe Kundenzufriedenheit
- “Schlechte” Qualität (viele Bugs)
- Fixe Releasedaten & Scope
- These: Produkt Entwicklung zu langsam?

Primäre Unternehmensziele

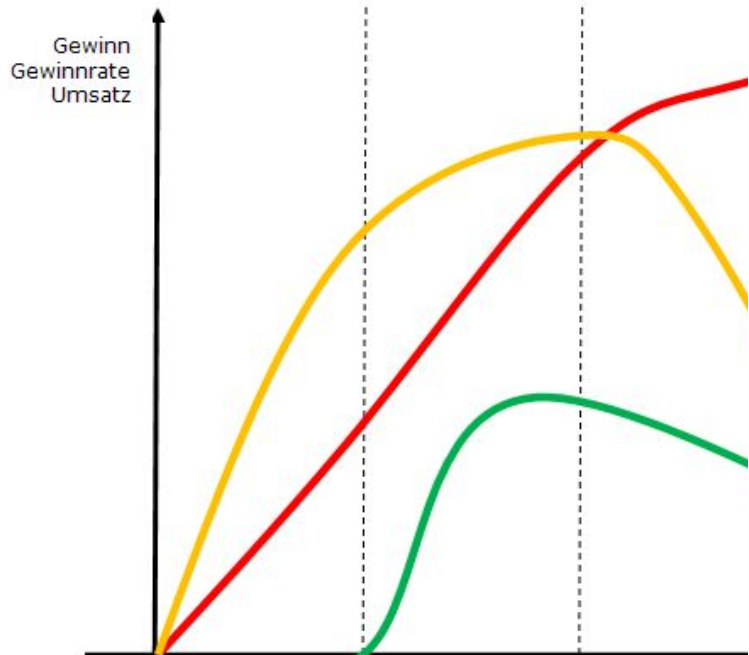
- Höhere Qualität
- Mehr Features
- Öftere Releases

Idee: Produktivität und Effizienz mit Scrum steigern



Produkt

Einordnung im Produktlebenszyklus



W
E
T
T
B
E
R
B
S
P
O
S
I
T
I
O
N

dominant
stark
günstig
haltbar
schwach

	Entstehung	Wachstum	Reife	Alter
dominant	-Marktanteile hinzugewinnen oder mindestens halten	-Investieren, um Position zu verbessern -Marktanteil-gewinnung	-Position halten -Wachstum mit der Branche	-Position halten
stark	-Investieren, um Position zu verbessern -Marktanteil-gewinnung (intensiv)	-Investieren, um Position zu verbessern -Marktanteil-gewinnung	-Position halten -Wachstum mit der Branche	-Position halten oder "Ernten"
günstig	-Selektive oder volle Marktanteil-gewinnung -Selektive Verbesserung der Wettbe-werbsposition	-Versuchsweise Position verbessern -Selektive Marktanteil-gewinnung	-Minimale Investi-tionen zur Haltung	"Ernten" oder stufenweise Reduzierung des Engagements
haltbar	-Selektive Verbes-erung der Wettbe-werbsposition	-Aufsuchen und Erhaltung einer Nische	-Aufsuchen einer Nische oder stufenweise Reduzierung des Engagements	-Stufenweise Reduzierung des Engagements oder -Liquidieren
schwach	-Starke Verbesserung oder aufhören	-Starke Verbesserung oder		



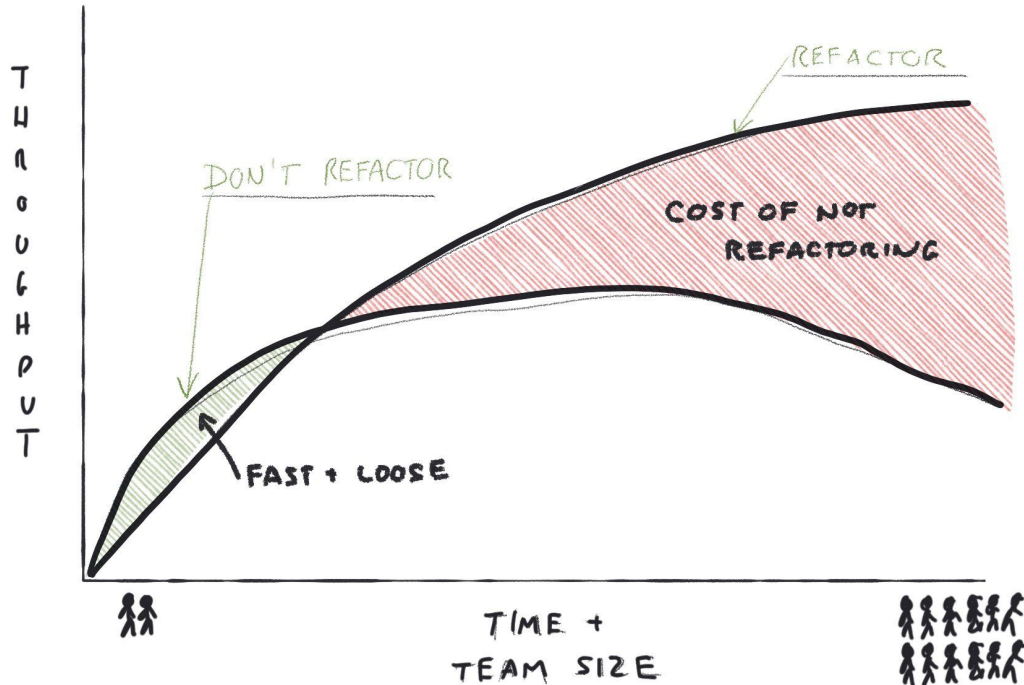
Unterschiedliche Vorstellungen und Handlungen

- Keine konkrete Produktstrategie und Mission
- Wechselhafte Produktstrategien

Anti Pattern bei Produkt

- Zu wenig Berücksichtigung von Kunden Feedback & Bewertung
 - Wir wissen was, der Kunde braucht. (“80% der Kunden wollen das.”)
 - Feature F***ing, Low hanging fruits, Quick wins etc.
 - Handeln: Dauer von Dev bis Code beim Kunden ca. 3 Monate
 - Zuhören: 4 Releases/ Jahr nicht vom Kunde gewollt
 - Lernen was der Kunde benötigt: Nutzen und Aufwand nicht angegangen
 - Kein Product Discovery
 - Effizienz: Output & Kostengetrieben > Effektivität: Outcome & Wertgetrieben
- Product Owner ist nicht ermächtigt
 - Priorisierung der großen strategischen Blöcke durch Management
 - Product Leads und Ausrichtung in Bereiche

Monolith & technische Schulden



Erste Zeile Code von 1992

C, C++, C#, Java, SQL,
Microsoft Technologien

Vermischung zwischen
GUI & Funktionalität

Refactoring vs neue Features

Anti Pattern bei der Entwicklung

“Legacy Code” Monolith vs. Modularisierung nicht als Problem angesehen

Lokale und nicht ganzheitliche Optimierung

- Teaming nach Komponenten
- Keine Modularisierungsstrategie / Architektur
- Viel manuelles Testing und sehr wenig automatisierte Tests

Durchschnittliche Teamkapazitäten werden in Features investiert
(Feature 70%; Bugfixing 20%; CIP 10%)

Weiterer Aufbau von Technical Debt und geringe Entwicklungsgeschwindigkeit

Prozess & Organisation

Neue Hierarchien & Organisation

- Management
 - CTO = Linien Manager
 - C-Level top down Ansatz
 - Agile Transition Team (ATT) = Management + ausgewählte Mitarbeiter + externer Coach
- Product Areas
 - Product Area Lead pro Produkt
 - Area Scrum Master
- Teams
 - Scrum Master = Ehemalige Team- & Projektleiter + 1 Dev
 - Product Owner = Ehemalige Product Manager + 1 neuer Mitarbeiter
 - Devs (z.B. DEV, QA, TW etc.)
 - Architekten in Teams, allerdings in Sonderrolle
 - CIP Team als Service Team für Dev
- Community of Practises (COPs) als Querschnitt Funktion

Anti Pattern bei der Organisation

Scrum Einführung nur Top-down, um Unternehmensziele zu erreichen

Teilzeit Agile Coaching und Abkündigung nach einem Jahr;
keine Weiterführung & Ermächtigung interner Mitarbeiter

Agile Transition ist “abgeschlossen” (nach ca. drei Jahren)

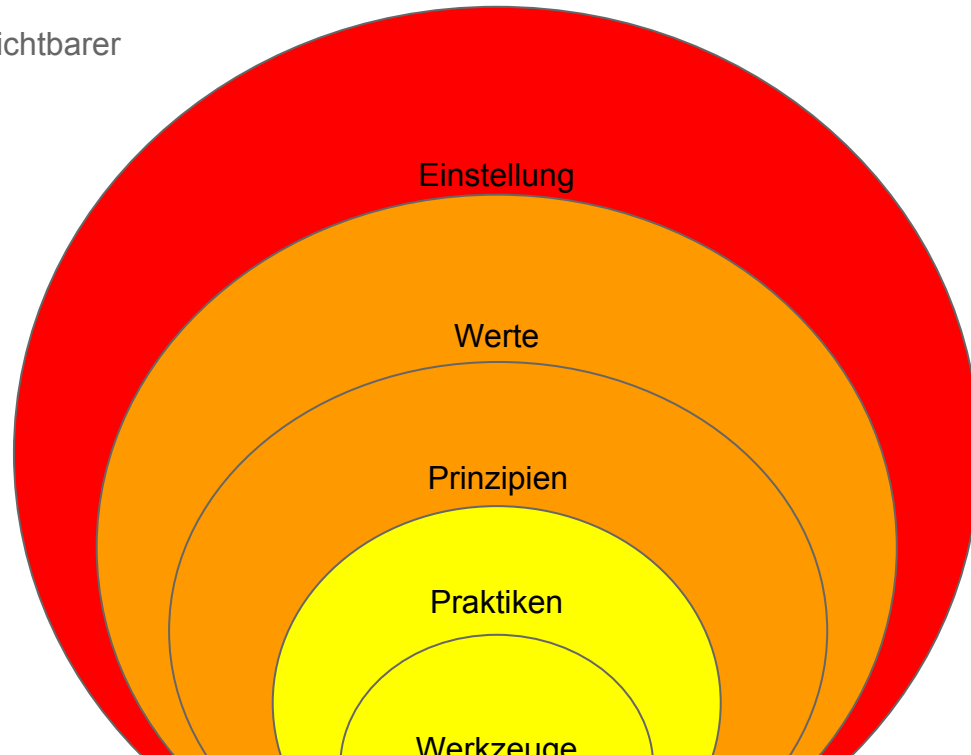
Keine Organisationsstrategie für Unternehmen - beschränkt auf
Entwicklungsbereich

Agile Transition auf Scrum beschränken und unterschiedliche Zielbilder zulassen

- Rahmen für Rollen und Handlungsspielraum nicht festlegen
- Kontinuierliche Überprüfung und Ausrichtung notwendig

Agile Transition vs. iterative SW Entwicklung mit Scrum

Wirkungsvoller und unsichtbarer



Entwicklung zur lernenden Organisation

Erfordert strukturellen und kulturellen Wandel

Kann in "Command & Control"-Kulturen

Eine nachhaltige Veränderung wird nur über die Veränderung der Einstellung erreicht.

Sichtbarer und wirkungsloser

Menschen

Anti Pattern auf Mitarbeitererebene I

Umgang mit ehemaligen Team- und Abteilungsleitern

Umstrukturierung ohne Klärung der Führungsaufgaben

360 Grad Feedback als Performance Messungen nutzen

Personalabteilung kaum involvieren & nicht in Agile Transition berücksichtigen

Personalentwicklung vernachlässigen

(Coaching, Gehalt, Perspektive, fachliche Weiterbildung)

Anti Pattern auf Mitarbeitererebene II

Hierarchien, Kastendenken und Kleinstgruppen dominieren;
Vermischung von Titel, Rollen und Status sowie fehlende Ermächtigung

Push in Selbstorganisation ohne kontinuierliche Förderung, Training & Führung

- Wettbewerb unter den Mitarbeitern (Awards, Wettbewerb etc.)
- Teams und Menschen alleine lassen
- Konflikte ignorieren

Umgang mit Menschen, die nicht agil werden wollen

FAZIT

Resümee

1. Verständnis von agiler Transformation formulieren
 - Agile Transformation != iterative Software Entwicklung mit Scrum
 - Bedeutung von Agilität im Unternehmenskontext
 - Handlungsspielräume von Rollen konkretisieren und transparent machen
2. Kultur folgt Struktur*: Prinzipien und Werte positiv Vorleben (Vorbildfunktion)
3. Inspect & Adapt: Kontinuierliche Verbesserung und Anpassung
 - Wertschätzung des Erreichten
 - Externer Austausch

Danke für die Aufmerksamkeit!



Stefan Görgner (@klaaaatuuuu) | Frank Gollas (@frankgollas) | Patrick Sauerwein (@virtual_patrick) | Marcus Schreiber (@MarcusSchreiber4)