Verträge in Agilen Softwareprojekten

„Vertrauen & Kooperation“

Disclaimer: keine Rechtsberatung!

Björn Schotte // @BjoernSchotte // bjoern.schotte@mayflower.de
MAYFLOWER GmbH

http://creativecommons.org/licenses/by-sa/3.0/deed.de

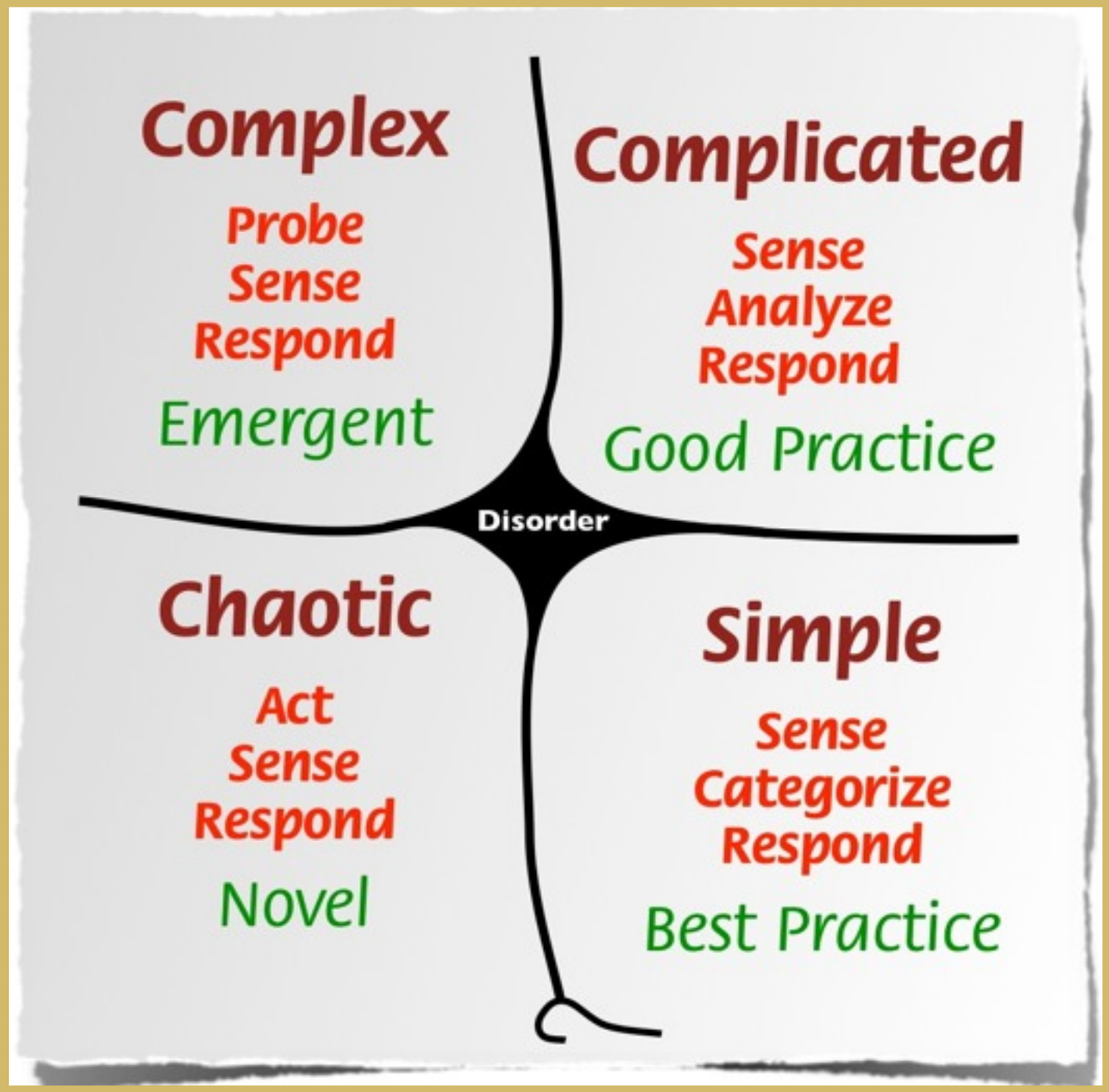
Über den Referenten: Björn Schotte

- Unruhestifter ;-)
- Geschäftsführer MAYFLOWER GmbH
- Senior Consultant
- hilft Kunden, die Herausforderungen ihres Geschäfts im Online Umfeld zu lösen. Berät konzeptionell & im Umfeld Agiler Methoden (Scrum, Enterprise Lean Startup)
- MAYFLOWER: 60+ Devs, Individualsoftware Web, Mobile & E-Business/E-Commerce

Ausgangslage Software-Entwicklung

- Cynefin (/ˈkʌnɨvɪn/) Modell

Softwareentwicklung:
Complex & Chaotic



Antwort auf Komplex & Chaotisch

- Agiles Projektvorgehen, emergente Praktiken
- Ich weiss heute nicht genau, was morgen sein wird
- Ich setze enge Korridore (Sprints), um kontinuierlich kleine Pläne „auf Sicht“ zu schmieden

Antwort auf Komplex & Chaotisch

- Upfront Design nur so viel wie notwendig und möglich
- … denn ich will gerade Flexibilität im Vorgehen haben
- Lasten-/Pflichtenheft: Irrglaube, die Dinge „im Griff“ zu haben, teure CRs, sehr viel höhere TCO

Ausgangslage Vertragswesen

- Verträge versuchen, Bekanntes zu regeln
- Verträge versuchen, Sicherheit zu bieten
- Verträge versuchen, Risiken zu verteilen
- Verträge versuchen, Lösungen für Probleme zu liefern
- Verträge werden (gemeinhin) dann genutzt, wenn sich die Parteien nicht mehr vertragen

Ausgangslage Vertragswesen

- Kunden haben oftmals schlechte Erfahrungen mit Dienstleistern gemacht und versuchen sich durch Verträge abzusichern (Risiko komplett auf Dienstleister abwälzen)
- dt. Werkvertragsrecht aus einer Zeit, als es noch keine Softwareentwicklung gab
- … was heisst denn eigentlich „Werkvertrag“?

Ausgangslage Vertragswesen

Definition Werkvertrag, Wikipedia:

„der Werkunternehmer schuldet dem Werkbesteller die Herstellung eines Werkes, das heißt die Herbeiführung eines bestimmten Erfolges tatsächlicher Natur.“

„Die Fälligkeit der Vergütung des Werkvertrages tritt mit der Abnahme des Werkes ein.“

Ähm …

Softwareentwicklung ...

KOMPLEX ...

Ich habe ein Problem, für das ich die Lösung noch nicht genau kenne

!= Werkvertrag

Hey, lass uns doch einfach T&M machen!

Pures T&M = Risiko 100%

auf Auftraggeberseite

Conclusio?

1.

Verträge eignen sich scheinbar nicht für agile Vorgehensweisen

2.
Software-Entwicklung
=
„für (un)bekannte Probleme mit noch unbekannten Lösungen“ (Scrum)

Komplex!

3.
Software-Entwicklung
=
„für unbekannte Probleme mit noch unbekannten Lösungen“ (Lean Startup)

Chaotisch!

Drama, Baby?

Annäherung, Teil 1

GPL, GNU General Public License

Kooperation durch Tit-for-Tat

Nutze den Source, verändere ihn. Das ist okay.

Verteilst du die neue Software,
liefere den gesamten Quellcode
mit. Inklusive deiner Änderungen.

Verhinderung von Missbrauch durch Hack

des Vertrages (Lizenz).

Erzwingt Kooperation.

Übertragbar auf unsere Problematik?

Annäherung, Teil 2

„missbrauche“ Vertragsrecht zum Kooperationszwang

Zweck des Vertrages wandelt sich

weg von Definition von Bekanntem
(wir wissen es eh nicht 100%ig)
hin zur Definition Rahmen, der
Kooperation regelt und Verletzung
von Kooperation ahndet

Auch im Vertrag Konzentration auf die Stärken agilen Vorgehens

Emergente Erkenntnisse nutzen, um flexibel am Markt zu agieren

Konzentration auf Generierung von hohem Business Value

Rahmen gestalten, der Kooperation ermöglicht und Pflichten/Rechte definiert

Mangelnde Kooperation ahnden

Nein, nicht mit Pönalen. Es gibt viel schlauere Lösungen.

Conclusio?

1.

Vertragsgestaltung „hacken“

2.

Konzentration auf Kooperation und Generierung von Nutzen

3.

Realität anerkennen. Empirie & emergentes Wissen zu Lösungen entsteht während des Projekts

Ergebnis:

Auflösung Unvereinbarkeit Agiles Vorgehen - Vertragsrecht

Beispiele für Vorgehens-/ Vertragsmodelle

Disclaimer - für alles gilt:

- fragen Sie Ihren Anwalt
- fragen Sie Mayflower :-)
- have fun & viele erfolgreiche Projekte!

T&M - the „good old one“

- bei klassisch T&M Bezahlung jeder Stunde (zB Sprint-weise)
- Risiko 100% auf Auftraggeber-Seite
- ohne weitere Elemente (zB enges Scrum, hohe Motivation etc)
  wenig Anreiz für den Dienstleister, hohen BV zu liefern
- dennoch: je nach Situation, Klarheit von Anforderungen, Möglichkeiten des Kunden kann pures T&M Sinn ergeben

„T&M mit Abbruch“

- bei klassisch T&M Bezahlung jeder Stunde (zB Sprint-weise)
- Kunde merkt nach N Sprints, dass nicht mehr viel Business Value zu holen ist
- mit einem Vorlauf von zB 1 Sprint kann der Kunde jederzeit nach einem Sprint das Projekt beenden
- Vorlauf notwendig, damit der Dienstleister die Ressourcen umplanen kann

„T&M mit Abbruch“: Beispiel

- Projekt ursprünglich auf 10 Sprints geplant
- gegen Ende Sprint 6 stellt der Kunde fest, dass kaum mehr BV zu holen ist, da sich die Marktbedingungen geändert haben
- Ende Sprint 6: Ankündigung, dass das Projekt mit dem Ende von Sprint 7 beendet wird
- Vorteile für beide Seiten, Kooperation wird gestützt:
  - kein BV mehr realisierbar? Abbruchmöglichkeit
  - DL hat genügend Vorlauf zur Umplanung

T&M on steroids

Warnung: nur für echte Männer

Regel 1:

T&M, sprint-weise Abrechnung aller Stunden

Regel 2:
Kunde kann ohne Angabe von Gründen einen Sprint nicht bezahlen

Steroids!

Regel 3:
Sonderkündigungsrecht DL, wenn der Kunde das 2. Mal einen Sprint nicht bezahlt

Ergebnis:

Verkettung beider Seiten in Kooperation

der Kunde überlegt sich sehr genau, wann er einen Sprint nicht bezahlt

der Dienstleister hat ein Interesse an guter, ergebnisorientierter Arbeit

Achtung: macht nur Sinn für Projekte, bei denen DL-Wechsel sehr schmerzhaft (teuer) ist und somit eine Garantie zur Kooperation benötigt wird

MAYFLOWER nutzt „T&M on steroids“ erfolgreich in Projekten.

Na, schon Herzrasen bekommen?

Ok, schalten wir einen Gang zurück.

Vertrag, klassisch, mit Gewerk und Jedöns...

Klassischer Vertrag, Agil mit Festpreis

- Regel 1: im Vertrag ist Scrum-Vorgehen definiert
- Regel 2: beiden Parteien bewusst, dass kein fixes Feature-Set geliefert werden kann. Das Budget ist fix definiert.
- Regel 3: es kann kein Gesamtwerk definiert werden

„Ein Gewerk brauche ich für Gewährleistung. Ich brauche Sicherheit. Hilfe, was kann ich tun?“

Klassischer Vertrag, Agil mit Festpreis

- Regel 4 (!): eine einzelne User Story stellt ein Mini-Gewerk da, auf das Gewährleistungsregeln angewandt werden
- Regel 5 (!): das Team kann User Stories ablehnen, wenn diese aus ihrer Sicht nicht ausreichend spezifiziert/schätzbar sind („können wir das Werkvertragsrisiko hier übernehmen?“)
- Regel 5 (! Variation): nach beidseitiger Rücksprache kann eine einzelne User Story dienstvertraglich behandelt werden

„Puh… ich hab Gewerk drin. Check. Doch wie ist das mit der Abnahme?“

Klassischer Vertrag, Agil mit Festpreis

- Regel 6 (!): Abnahme des Mini-Gewerks im Sprint Review. Bei Nicht-Abnahme (weil Story nicht errreicht!) kommt die Story zurück ins Backlog und wird zB für den nächsten Sprint wieder eingeplant.
- Regel 7 (!): monatlicher Zahlungsplan aufgrund der erbrachten Leistungen
- Regel 8 (!): wird eine bereits realisierte User Story durch eine neue US erweitert/ersetzt, so beginnt die Gewährleistung neu

„Ok. Entkopplung Bezahlung von Gewährleistung/Abnahme. Nur wenn im Nachhinein, also im Betrieb, etwas Abgenommenes nachweislich nicht stimmt, muss kostenlos gefixt werden. Das ist fair.“

Kooperation auf beiden Seiten.

Vorgehen im Vertrag festgelegt.

Mini-Gewerke statt großes Gewerk.

Stakes von Legal & Procurement erfüllbar.

MAYFLOWER nutzt „Klassisch-agiler Festpreis“ erfolgreich in Projekten.

Back to Innovation:

Money for Nothing, Changes for Free

Exkurs zurück ...

Verträge, old school:

„Ich kenne die Features, ich kenne den ROI, ich kann alles definieren und regeln.“

Das stimmt so nicht

Daher Agile Logik:

Ich investiere (und bezahle) Arbeitszeit

Ich erzeuge maximalen Business Value

und mache so lange

weiter

bis es sich nicht mehr lohnt

(Kosten Feature >> BV)

=

bis es sich nicht mehr lohnt,
weiter zu machen

Ausprägung dieses Prinzips ist „Money for nothing, Changes for free“

Regel 1:

Standard fixed price Contract mit T&M für Changes

Regel 2:
Tausch von Features gleicher SP Zahl möglich, sofern an US noch nicht begonnen (Changes for Free)

Regel 2a:
Neue Features möglich, wenn Low Prio Features gleicher SP-Zahl wegfallen

Anforderung an Kunde & DL: es gibt ein qualitativ gutes Backlog, Gesamt-Features sind gut schätzbar!

Regel 3:
hält der Kunde sich nicht an Scrum, wandelt sich das Projekt in reines T&M

Nun zum „Money for Nothing“ …

Regel 4:
ebenfalls nur gültig, wenn Scrum Vorgehen eingehalten

Regel 5:
beide Parteien können sich auf SP Estimates einigen.
Ansonsten Umwandlung in T&M

Regel 6:
wenn
Kosten Feature >> BV
dann Projekt-Abbruch

Regel 7:
Ausgleich für frühzeitige Beendigung des Projekts: 20% des übrig gebliebenen Budgets geht zusätzlich an DL („Money for Nothing“)

Hmm.

Wann ist der Einsatz

von M4NC4F sinnvoll?

Nur bei BV Optimierung!
Nur wenn klar ist, dass ich das Budget nicht bis zum Ende aufbrauchen will.

M4NC4F lohnt sich nicht,
wenn das Budget sowieso
gut und sinnvoll genutzt
werden kann.

(iSv gute Business Value Generierung regelmäßig möglich)

MAYFLOWER nutzt M4NC4F erfolgreich in Projekten.

Finale …

ein paar Empfehlungen

Konzentrieren Sie sich auf vertrauensvolles Arbeiten auf Augenhöhe

Setzen Sie im Vertrag nur einen Rahmen,
der Kooperation garantiert und
Verletzung von Kooperation ahndet.

Typische aus Legal & Procurement getriebene Pönalen sind tabu. Sie ergeben im agilen Kontext keinen Sinn.

Ahnden Sie Verletzung der Kooperation mit Abbruchmöglichkeiten (= Hebel für Auftraggeber) oder T&M Wandlung (= Hebel für Auftragnehmer)

Legal & Procurement müssen mit ins Boot. Und intensiv beraten werden.

Noch ein Tipp für Ihre Feature-Planung.

Die User Story muss den Nachweis erbringen, dass der Wert erwirtschaftet wird

Es wird der erwartete Wert angehangen.

~~(Business Value Poker)~~

Ich beschreibe den Test, was ich mir durch dieses Feature erwarte

(zB 5% mehr Conversion Rate)

Nach Release validiere ich meine Erwartung.

Lerne & Adaptiere.

Liege ich regelmäßig mit meinen Erwartungen daneben? Hmm. Dann werden meine Feature-Wünsche nicht mehr berücksichtigt.

Buch- und Linktipps

Buch „Der Agile Festpreis“ (Boris Gloger) mit weiteren Ideen

Google-Suche zu „10 contracts for your next agile project“

„Lasst uns agile Projekte besser machen. Durch gegenseitiges Vertrauen & Kooperation“

MAYFLOWER

http://mayflower.de/

CC BY SA