

Europa Institut an der Universität Zürich  
ICT – Recht und Praxis, 14. September 2016

## **Dynamischere Anforderungen – Agilere Verträge?**

Wie können Verträge Komplexität und Geschwindigkeit von ICT-Projekten unterstützen?

Reto Maduz  
Wolfgang Straub



**Es war einmal ein  
Software-Entwicklungsprojekt...**

## ...in mehreren Akten




3




- > Neuentwicklung einer Tablet Applikation für Verkauf von Versicherungslösungen
- > Anforderungen: Alles was auf Papier geht - nur moderner
- > Vergabe an externen Partner da Inhouse-Wissen beschränkt
- > Man entscheidet sich, gemeinsam „agil“ mit Scrum vorzugehen

## Übersicht

- > Vorgehensmodelle
- > Projektinitialisierung
- > Umgang mit neuen Anforderungen
- > Realisierungsschwierigkeiten
- > Abnahme und Übergabe an Betrieb
- > Lessons Learned

Generelle Informationen 

Fallbeispiel 

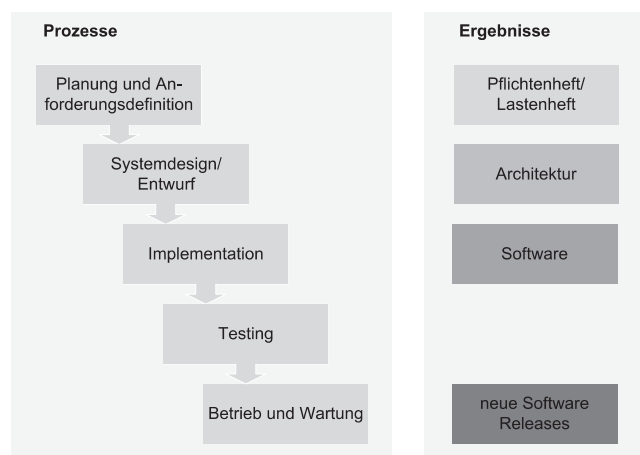
Lösungsansätze 

5

## Vorgehensmodelle



Wie man traditionellerweise vorgegangen wäre...



6

## Vorgehensmodelle



### Probleme klassischer Vorgehensmodelle

- > Im Zeitpunkt der Spezifikation weiss man oft noch zu wenig
- > Changes führen zu Mehraufwand
- > Realisierungsfortschritte sind schwierig zu beurteilen
- > Nutzen entsteht nach Ende der Implementierung
- > Viel, was während Spezifikation als wichtig erachtet wurde, wird später nach Realisierung nie gebraucht.

7

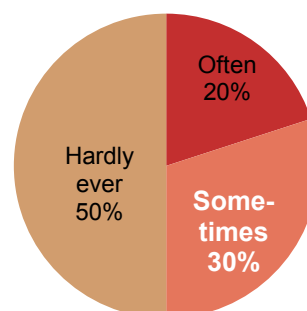
## Vorgehensmodelle



### Probleme klassischer Vorgehensmodelle

- > Oft wird das Produkt entwickelt das bestellt wurde und nicht das, welches gebraucht wird.
- > Nur 20% der Funktionalitäten werden regelmässig verwendet; 50% kaum oder nie.

### Features used



Source: Standish Group  
Chaos Manifesto 2013

8

## Vorgehensmodelle



### Agile in der Praxis

- > In der Schweiz erfolgten 2015 **ca. 44% der Softwareentwicklungen agil**
- > **Scrum** ist das weitaus verbreitetste Vorgehensmodell (über 84% der agilen Entwicklungen)
- > In der Praxis kommen aber oft **hybride Mischformen** vor (z.B. Hermes Entwicklung Agil, Bimodale Methoden)

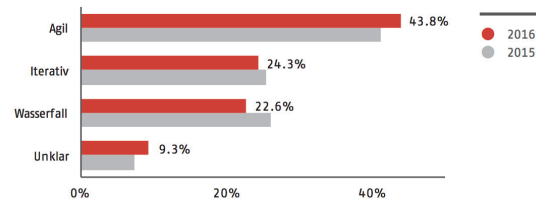
Quelle: SWISSQ (Hrsg.), Software Development 2016: Agile, Requirements, Testing; www.swissq.it

9

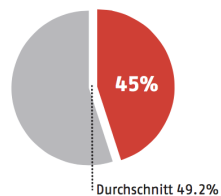
## Vorgehensmodelle



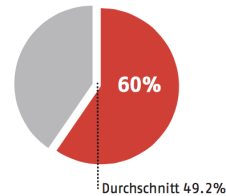
### Vorwiegendes Vorgehen im Projekt



### Projekterfolg Wasserfall



### Projekterfolg Agile



Quelle: SWISSQ (Hrsg.), Software Development 2016: Agile, Requirements, Testing; www.swissq.it 10

## Vorgehensmodelle



### Scrum Kernelemente

- > Initialisierung mit **Product Vision**
- > Beschreibung der einzelnen Funktionen als **User Stories**, aber **keine abschliessende Spezifikationen**
- > Regelmässige Repriorisierung der User Stories für Umsetzung
- > Implementierung in **Sprints** von 2-4 Wochen
- > Auslieferung und Testing von **Product Increments** nach jedem Sprint
- > Kontinuierliches **Feedback** vom Auftraggeber
- > Eventuell Zusammenfassung mehrerer Product Increments zu **Releases**

11

## Projektinitialisierung



Anforderungen fallen nicht vom Himmel...

12

## Projektinitialisierung



Auch bei agilen Projekten ist eine Initialisierungsphase notwendig:

- > **Erarbeitung Product Backlog**
  - Funktionale Anforderungen (User Stories)
  - Nichtfunktionale **Anforderungen**
  - **Priorisierung** der Anforderungen
  - Schätzung Realisierungsaufwand/Nutzwert
- > **Architektur Design**
  - Struktur der Software, Modularität, Schnittstellen etc.
- > **Klärung der Projektorganisation, etc.**
- > ...

13

## Projektinitialisierung



### Vertragsstrukturen

- > **Vertrag über Initialisierungsphase**/Letter of Intent, Danach Entscheid über Realisierung
- > **Gesamtvertrag**
- > **Rahmenvertrag** mit gemeinsamen Bedingungen für Initialisierung und Realisierung
  - Eventuell Einzelvertrag über Initialisierungsphase
  - Einzelverträge für Sprints/Releases
  - Wartungs- und Supportvertrag

14

## Projektinitialisierung



### Anwendung auftragsrechtlicher oder werkvertraglicher Regeln?

- > 'Werk oder Wirken?'
- > Wer trägt Entwicklungsrisiken/Ergebnisverantwortung?
- > Vergütungsmodell (ist nur ein Indiz!)
- > Kombinationen (z.B. auftragsrechtlicher Rahmen und werkvertragliche Sprints)
- > Eventuell nur analoge Anwendung der einen oder anderen Regeln (Innominatkontrakte)

15

## Projektinitialisierung



Wie wird die Zusammenarbeit aufgesetzt?

### → Vertragliche Rahmenbedingungen

- > Projektorganisation
- > Prozesse (z.B. Repriorisierung von User Stories, Austausch von Mitarbeitenden, Einforderung von Mitwirkungspflichten)
- > Vergütungsmodell (pro Sprint, Release? nach Aufwand? Bonus?)
- > Risiko- und Chancenverteilung
- > Rechte an Arbeitsergebnissen/Immaterialgüterrechten
- > Abnahmeverfahren
- > Gewährleistungsmodalitäten
- > Zeitrahmen (Exit Points, Beendigungsmodalitäten)

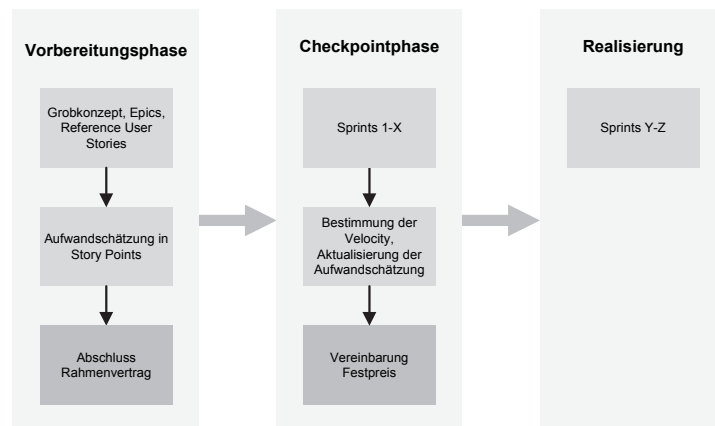
16



## Projektinitialisierung



### 'Agiler Festpreis'



17

## Projektinitialisierung



### 'Agiler Festpreis'

- > **Projektscope** sowie Reference **User Stories** werden definiert
- > Komplexität wird nach Entwicklungsaufwand und Risiken mit **Storypoints** bewertet
- > In der **Checkpoint-Phase** wird die Entwicklungsgeschwindigkeit des Teams (**Velocity**) in 2-5 Sprints getestet
- > Am Ende der Checkpoint-Phase wird die **Schätzung aktualisiert und ein Festpreis vereinbart**

18



## Neue Anforderungen tauchen auf...

19



## Umgang mit neuen Anforderungen

Der **Product Backlog** verändert sich im Lauf der Realisierung:

- > **Repriorisierung** der User Stories
  - Ursprüngliche Anforderungen werden als nicht mehr relevant angesehen
  - Andere Anforderungen werden höher priorisiert
- > **Interpretationsfragen** von Anforderungen
- > **Neue Anforderungen** kommen in den Fokus
- > Der Gesamtaufwand verändert sich (z.B. 'Scope Creep' um 30%)

20

## Umgang mit neuen Anforderungen



### Lösungsansätze im Modell 'Agiler Festpreis':

- > Der Projektscope wird zum **Festpreis** realisiert
- > **Changes ausserhalb des Projektscope** gehen zulasten des Auftraggebers
- > **Changes innerhalb des Projektscope** sollen primär durch Reduktion/Verzicht auf bereits definierte User Stories ausgeglichen werden
- > Zusatzaufwand zwischen den Parteien wird nach voraus definierten Regeln aufgeteilt (z.B. **Riskshare**)

21



## Realisierungsschwierigkeiten und Prozessanpassungen

22

## Realisierungsschwierigkeiten...



'Unvorhergesehene' Probleme tauchen auf...

- > Die **Entwicklung** kommt **langsamer** voran, Gründe sind kontrovers
- > Der **Auftraggeber** ist plötzlich stark absorbiert im Daily Business
  - Er steht kaum mehr als Ansprechpartner zur Verfügung
  - Entscheidungen können nicht mehr rechtzeitig getroffen werden
- > **Projektplanung** wird angepasst
  - Hat dies auch Auswirkungen auf Vergütungen (z.B. Boni)?
- > Auftraggeber möchte nun auf **Wasserfallmodell** wechseln
  - Ist dies noch möglich?

23

## Realisierungsschwierigkeiten...



### Vertragliche Vorsorgemöglichkeiten

- > **Prozesse** definieren
  - Einforderung von Mitwirkungspflichten
  - Regeln zum Change Management (auch Änderungen vertraglicher Bestimmungen)
  - Voraussetzungen für Austausch von Mitarbeitenden
  - Konflikteskalationsverfahren
  - ...
- > **Incentives** (→ falsch konzipierte Incentives sind gefährlich!)
- > **Exit Möglichkeiten** (z.B. nach jedem Sprint)

24



## Abnahme und Übergabe an Betrieb

25



## Abnahme und Übergabe an Betrieb

- > Welche Bedeutung haben **Sprintabnahmen** (z.B. Vorabnahmen, Teilabnahmen)?
- > Was wird später noch abgenommen (z.B. Abnahme von Releases, Gesamtabnahme, Integrationsabnahme, User Acceptance Test)?
- > Gibt es **finanzielle Anreize** (z.B. Boni, Garantierückbehalte)?
- > Wie wird die **Übergabe an den Betrieb** gehandhabt (z.B. sequentiell pro Sprint, Big Bang)?
- > Wie sollen **Wartung und Support** gestaltet werden?

26

## Abnahme und Übergabe an Betrieb



### Abnahmekriterien

- > **Definition of Done**
  - **Qualitätskriterien** (z.B. Funktionalität, Einhaltung von Programmierrichtlinien, Antwortzeiten)
  - **Nichtfunktionale Anforderungen**
  - **Einschränkungen** (z.B. für bestimmte User Stories)
- > Eventuell **zusätzliche Akzeptanzkriterien** für bestimmte User Stories
- > Eventuell zusätzlich **Criteria of Satisfaction** (z.B. Voraussetzung für Prämien)

27

## Abnahme und Übergabe an Betrieb



### Gewährleistung – vertraglicher Regelungsbedarf

- > **Beginn** (z.B. Product Increments/Releases/Gesamtabnahme)?
- > Wiederaufleben durch **neue Softwareversionen**?
- > **Nachbesserungsmodalitäten** (z.B. im Rahmen nächster Sprints oder durch separate Patches/Releases?)
- > Verhältnis zu allfälligen **Wartungsvereinbarungen**?

28

## Lessons Learned



Wie können Verträge Komplexität und Geschwindigkeit von ICT-Projekten unterstützen?

- > Bedeutung der **Initialisierungsphase**
- > Aktive **Mitwirkung des Auftraggebers** (personelle Ressourcen, kurze Entscheidungswege etc.)
- > **Prozesse** durchdenken und im Vertrag verankern
- > **Gewährleistungsfragen** regeln
- > Betriebsphase: **Wartung und Support** von Anfang an planen

29

## Fragen, Anregungen, Kritik?

### Reto Maduz

SwissQ Consulting AG  
Fraumünsterstrasse 16  
CH-8001 Zürich

+41 43 288 88 40  
reto.maduz@swissq.it

www.swissq.it  
Twitter @rmaduz

### Wolfgang Straub

Augsburger Deutsch & Partner  
Effingerstrasse 17/Postfach  
3001 Bern

+41 31 381 37 15  
wolfgang.straub@ad-p.ch

www.ad-p.ch  
www.it-recht.ch

30