

Inhalt der Vorlesung

- Einführung
- Kommunikation
- Software Qualität
- Vorgehensmodelle
- Requirements Engineering
- Software Architektur und – Design
- Konfiguration Management

Vorgehensmodelle



- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

- Überblick (Wiederholung)

- Einleitung

- Wichtige Vorgehensmodelle – nicht agil

- Agile Methoden

- Agile Methoden allgemein

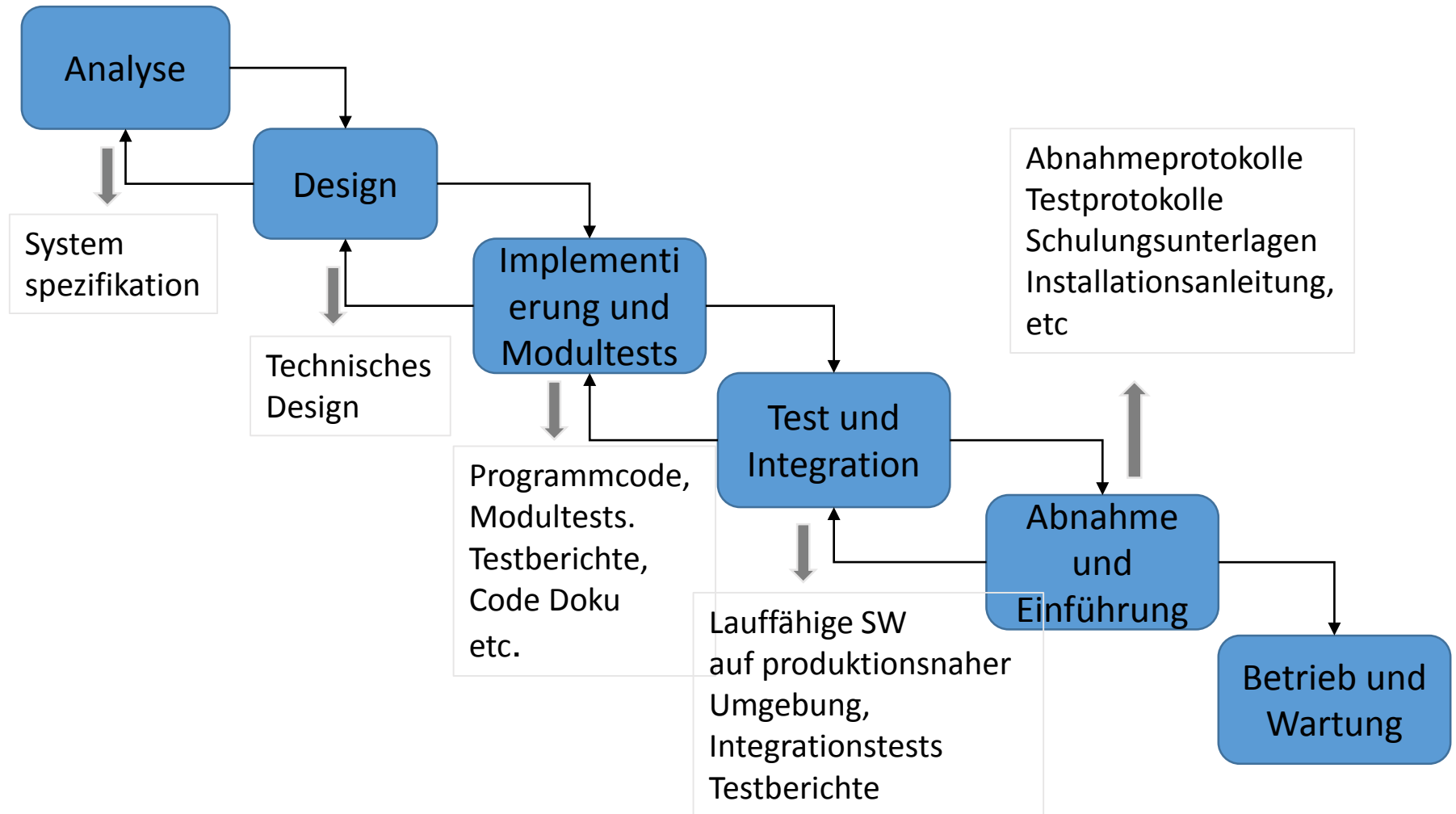
- Wiederholung: Scrum

- Scrum in großen Projekten

- Vertragsmodelle

- Agiler Festpreis

Tätigkeiten und Artefakte in einem SW Projekt



Ein Vorgehensmodell beschreibt:

- Menge von Aktivitäten
- Produkte
- Rollen
- Vor- und Nachbedingungen

Es gibt kein ideales und für alle Projekte passendes Vorgehensmodell.

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

- Code and Fix
- Wasserfallmodell
- Prototyping
- V- Modell
- Rational Unified Process (RUP)
- Spiralmodell
- Iterative Entwicklung
- Inkrementelle Entwicklung
- Agile Methoden → nächstes Kapitel

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

Kritik an klassischen Vorgehensmodellen

- Es müssen viele Dokumente erzeugt und gepflegt werden.
- Eigene Wissenschaft Modelle wie V-Modelle und RUP zu verstehen und zurecht zu schneiden.
- Prozessbeschreibungen hemmen Kreativität.

Alternativer Ansatz:

**Menschen machen Projekte erfolgreich, traue
den Menschen.**

=> agile Prozesse

Agile Software Entwicklung

VersionOne, the 10th annual state of agile report
(download unter
<http://www.versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>)

95% aller Unternehmen setzen agile Prozesse ein.

Agile Software Entwicklung

Agile Werte



Agiles Manifest

Agile
Prinzipien



Agile Prinzipien

Agile
Methoden



Pairprogramming
TDD
Refactoring
...

Agile
Prozesse



Scrum
Kanban
XP
...

Agiles Manifest (Februar 2001)

<http://www.agilemanifesto.org/iso/de/>

Wir erschließen bessere Wege, Software zu entwickeln,
indem wir es selbst tun und anderen dabei helfen.
Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- **Individuen und Interaktionen mehr als Prozesse und Werkzeuge**
- **Funktionierende Software mehr als umfassende Dokumentation**
- **Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung**
- **Reagieren auf Veränderung mehr als das Befolgen eines Plans**

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden,
schätzen wir die Werte auf der linken Seite höher ein.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler,
James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C.
Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas
www.agileAlliance.org

<http://agilemanifesto.org/iso/de/principles.html>

Prinzipien hinter dem Agilen Manifest

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.

6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit -- die Kunst, die Menge nicht getaner Arbeit zu maximieren -- ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

Zitat Andy Hunt:

“Agile methods ask practitioners to think, and frankly, that’s a hard sell.”

Aus: <http://blog.toolshed.com/2015/05/the-failure-of-agile.html>

(Beachten Sie den Titel!)

Was macht agile Methoden schwierig?

- Bedarf an Regeln
- Angst, sich zu blamieren
- „Save your ass“ Mentalität
- ...

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

Gliederung

- Einführung in Scrum
- Einschätzung
- Referenzen

Ansatz von Scrum: empirisch, inkrementell, iterativ

Prinzipien:

- Transparenz
- Überprüfung
- Anpassung

Idee: In kurzen Zyklen releasefähige Software auszuliefern

Rollen: Product Owner, Scrum Master, Entwicklungs Team, (*nach Gloger zusätzliche Rollen:* Customer, Manager, User)

Meetings: Sprint Planning (1&2), Daily Scrum, Sprint Review, Sprint Retrospective, (*nach Gloger zusätzlich:* Estimation Meeting)

Artefakte: Product Backlog, Sprint Backlog, Produkt Inkrement (*nach Gloger zusätzlich:* Vision, Sprint Goal, Tasks, Releaseplan, Impediment Backlog)

Definition of Done

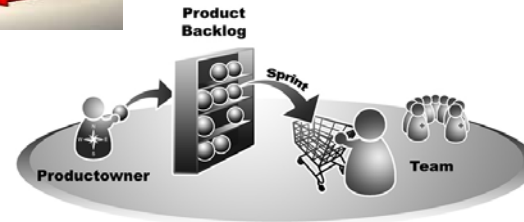
- Scrum Teams sind **selbstorganisierend** und **interdisziplinär**.
- Selbstorganisierende Teams entscheiden selbst, wie sie ihre Arbeit am besten erledigen, anstatt dieses durch andere Personen außerhalb des Teams vorgegeben zu bekommen.
- Interdisziplinäre Teams verfügen über alle Kompetenzen, die erforderlich sind, um die Arbeit zu erledigen, ohne dabei von Personen außerhalb des Entwicklungsteams abhängig zu sein.
- Das Team-Modell in Scrum wurde konzipiert, um **Flexibilität, Kreativität und Produktivität** zu optimieren.

Vier Arbeits-Prinzipien bei Scrum

- Selbstorganisation



- Pull Prinzip



- Timebox



- Nutzbare Funktionalität



Scrum – der Product Owner

Product Owner:

Der Product Owner ist für die Wertmaximierung des Produkts sowie die Arbeit des Entwicklungsteams verantwortlich. Wie dies geschieht, kann je nach Organisation, Scrum Team und Einzelpersonen stark variieren. Der Product Owner ist die einzige Person, die für das Management des Product Backlogs verantwortlich ist.

Scrum – das Entwicklungsteam

Das **Entwicklungsteam** besteht aus Profis, die am Ende eines jeden Sprints ein fertiges Inkrement übergeben, welches potentiell auslieferbar ist. Nur Mitglieder der Entwicklungsteams erstellen das Produkt-Inkrement.

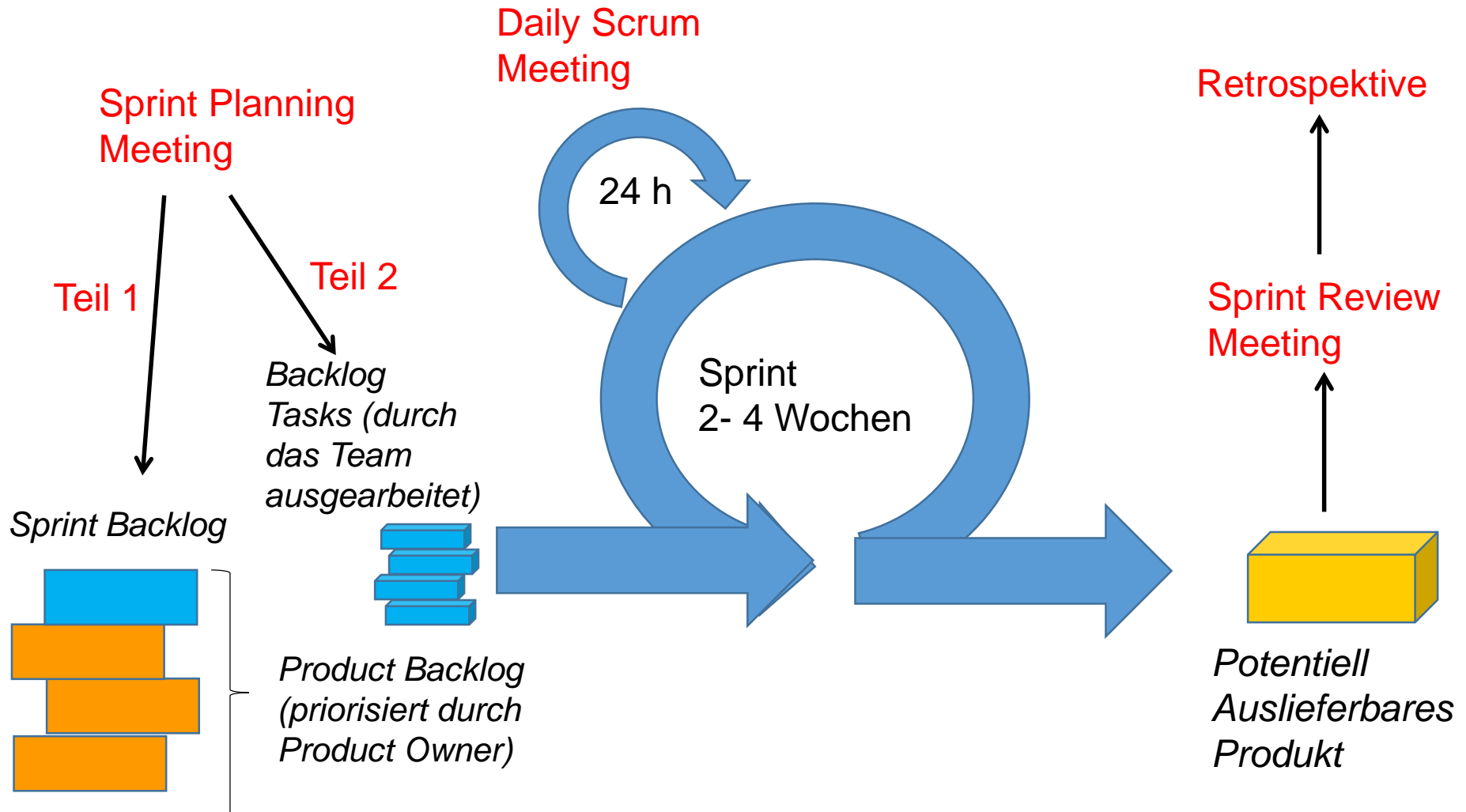
Scrum – der Scrummaster

Der **Scrum Master** ist für das Verständnis und die Durchführung von Scrum verantwortlich. Er tut dies, indem er dafür sorgt, dass das Scrum Team die Theorie, Praktiken und Regeln von Scrum einhält.

<https://www.youtube.com/watch?v=oheekef7oJk>

- **Customer**
Auftraggeber. Finanziert das Projekt.
- **Manager**
stellt Ressourcen und Richtlinien innerhalb der Organisation bereit.
- **User**
wesentliche Informationsquelle für das Scrum Team.

Scrum Prozess



Der Sprint als zentraler Container für die Arbeit:

Ein Sprint beinhaltet und umfasst das Sprint Planning, die Daily Scrums, die Entwicklungsarbeit, das Sprint Review und die Sprint Retrospektive.

Während des Sprints:

- werden keine Änderungen vorgenommen, die das Sprint-Ziel gefährden,
- wird der Qualitätsanspruch nicht geschmälert, und
- der Anforderungsumfang kann zwischen Product Owner und Entwicklungsteam geklärt und neu ausgehandelt werden, wenn sich neue Erkenntnisse ergeben haben.

Die Meetings - Sprint Planning

Im **Sprint Planning** (manchmal geteilt in zwei Meetings) wird die Arbeit für den kommenden Sprint geplant. Dieser Plan entsteht durch die gemeinschaftliche Arbeit des gesamten Scrum Teams.

Das Sprint Planning beantwortet die folgenden Fragen:

- Was ist in dem Produkt-Inkrement des kommenden Sprints enthalten?
- Wie wird die für die Lieferung des Produkt-Inkrementes erforderliche Arbeit erreicht?

Zweigeteiltes Sprint Planning

Sprint Planning 1:

entspricht einem **Anforderungs Workshop**,
Ergebnis: Selected Product Backlog (Sprint Backlog)

Sprint Planning 2:

Entspricht **Design Workshop**,

Ergebnis: Liste der Tasks für den Sprint am
Taskboard

Meetings - Daily Scrum

**Jedes Team
Mitglied
beantwortet drei
Fragen:**

1. Was habe ich
gestern gemacht?
2. Was werde ich
heute tun?
3. Was hindert mich
bei meiner
Arbeit?

Scrum Board



Meetings – Sprint Review

Am Ende eines Sprints wird ein **Sprint Review** abgehalten, um das [Produkt-]Inkrement zu überprüfen und das Product Backlog bei Bedarf anzupassen. Während des Sprint Reviews beschäftigen sich das Scrum Team und die Stakeholder gemeinsam mit den Ergebnissen des Sprints.

Meetings - Retrospektive

Die Sprint Retrospektive bietet dem Scrum Team die Gelegenheit, sich selbst zu überprüfen und einen Verbesserungsplan für den kommenden Sprint zu erstellen.

Estimation Meeting

Dieses Meeting ist nicht im Original Scrum vorgesehen. Es wird jedoch in vielen Projekten praktiziert.

- PO und Team schätzt mindestens einmal pro Sprint das Backlog, priorisiert neu und aktualisiert ggfs den Releaseplan.

Schätzen in Scrum

Aussage aus dem offiziellen Scrum Guide:

Ein Product Backlog-Eintrag enthält als Attribute eine Beschreibung, die Reihenfolge, die Schätzung und den Wert.

Keine Aussagen über die Art der Schätzung oder die Einheit des Schätzwerts.

Schätzen von Storypoints

Verschiedene Definitionen:

- Je höher die Anzahl der Storypoints, desto höher der Geschäftsnutzen der Userstory.
- Aufwand, Einheit jedoch nicht festgelegt, d.h. Aussagen nur relativ zu anderen Userstories möglich.
- Geschäftsnutzen/Aufwand – nicht wirklich definiert.
- Gloger: „Die Größe bezeichnet den Grad des Verständnisses, welches das Team von dem Backlog Item, von der Funktionalität hat.“

Skala häufig Fibonacci Zahlen.

Planning Poker

- *Einigung auf Referenz Backlog Item*

Dann:

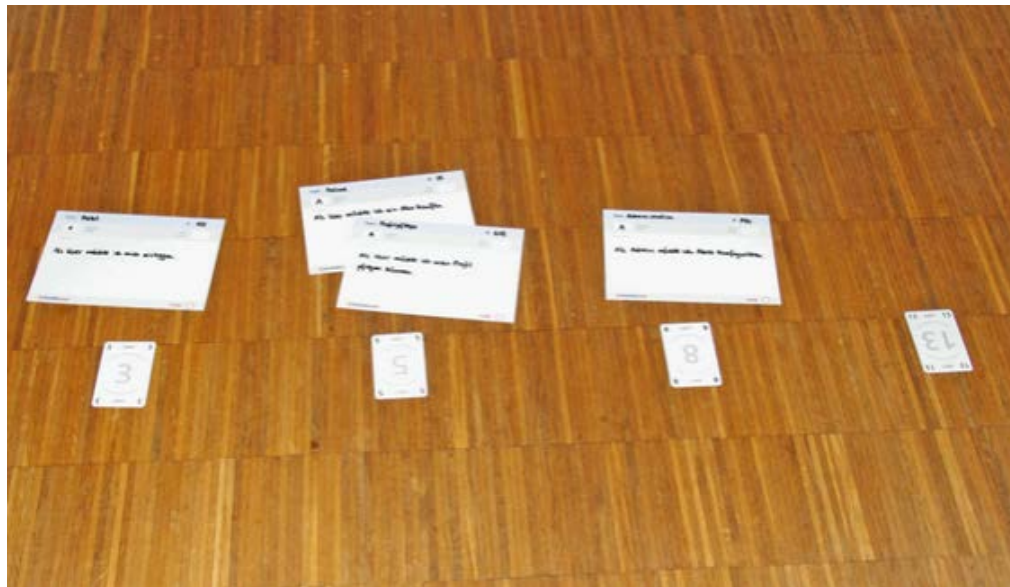
Für die zu schätzenden Backlog Items:

- *Vorstellung eines Backlog Items*
- *Klärung offener Fragen*
- *Teammitglieder vergeben Storypoints*
- *Diskussion*
- *Wiederholung bis ein Ergebnis vorliegt*



Magic Estimation

- Jedes Teammitglied erhält ausgedruckte BL Items.
- Die Items werden auf einer Skala angeordnet und dadurch geschätzt.
- Jedes Teammitglied kann die Items verschieben, bis die Lage stabil ist.



Aussage:

Schätzungen, die man für etwas brauchen kann, schätzen den Aufwand.

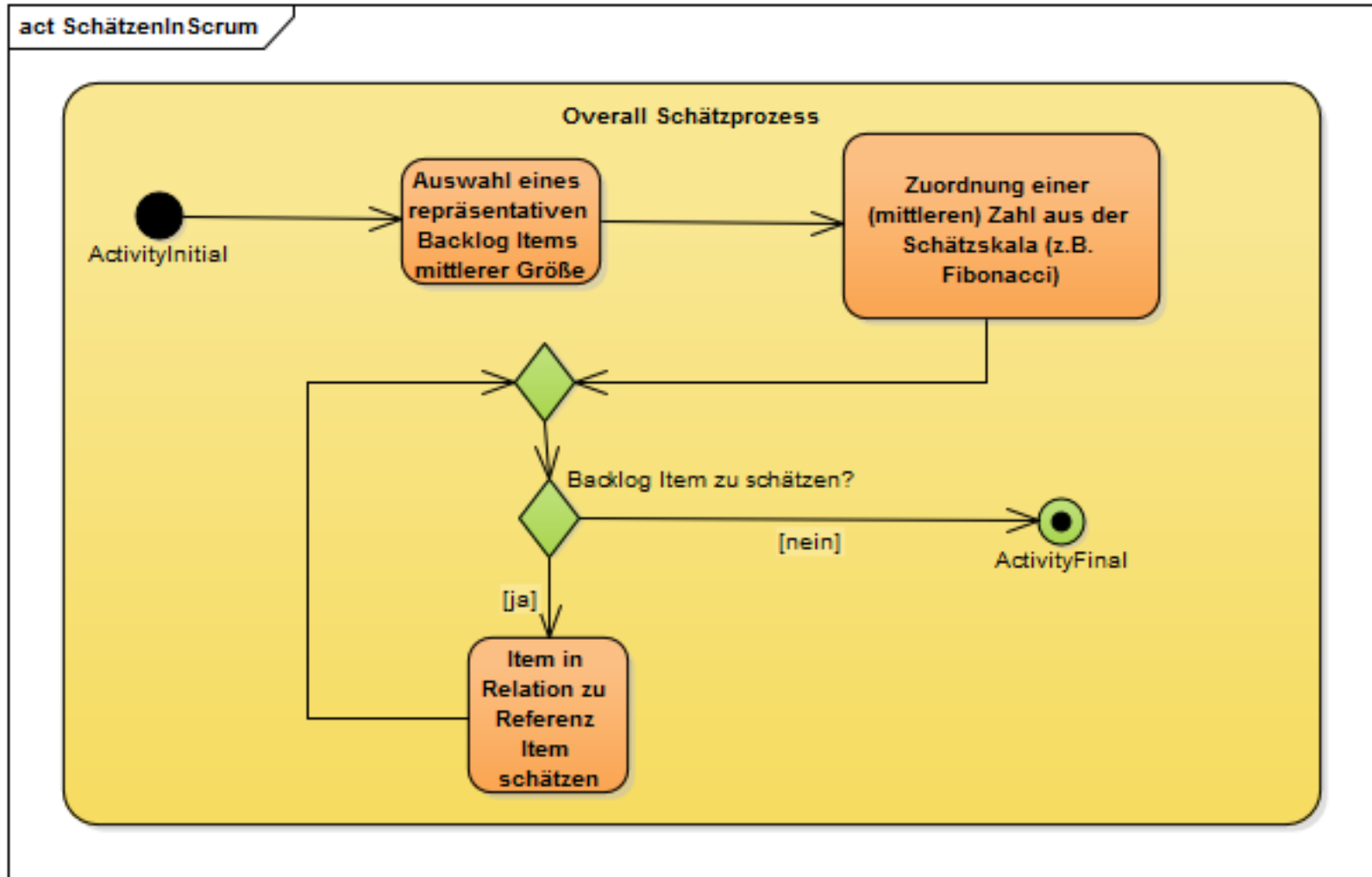
Erkenntnis:

Zuverlässiges Schätzen ist schwer bis unmöglich.

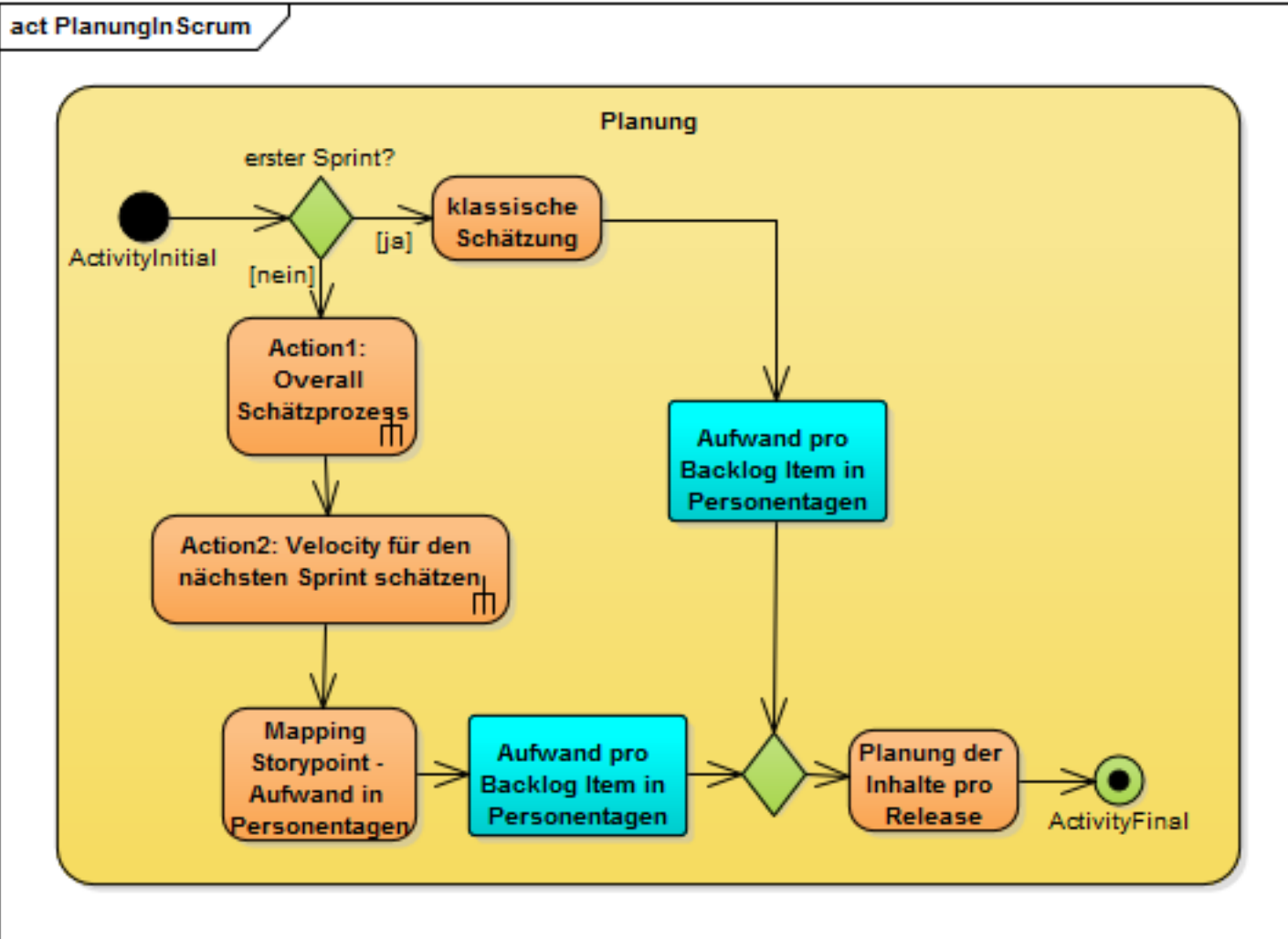
Also was tun?

➔ Es werden nur relative Aufwände geschätzt. Absolute Aufwände folgen aus der Erfahrung einiger Sprints.

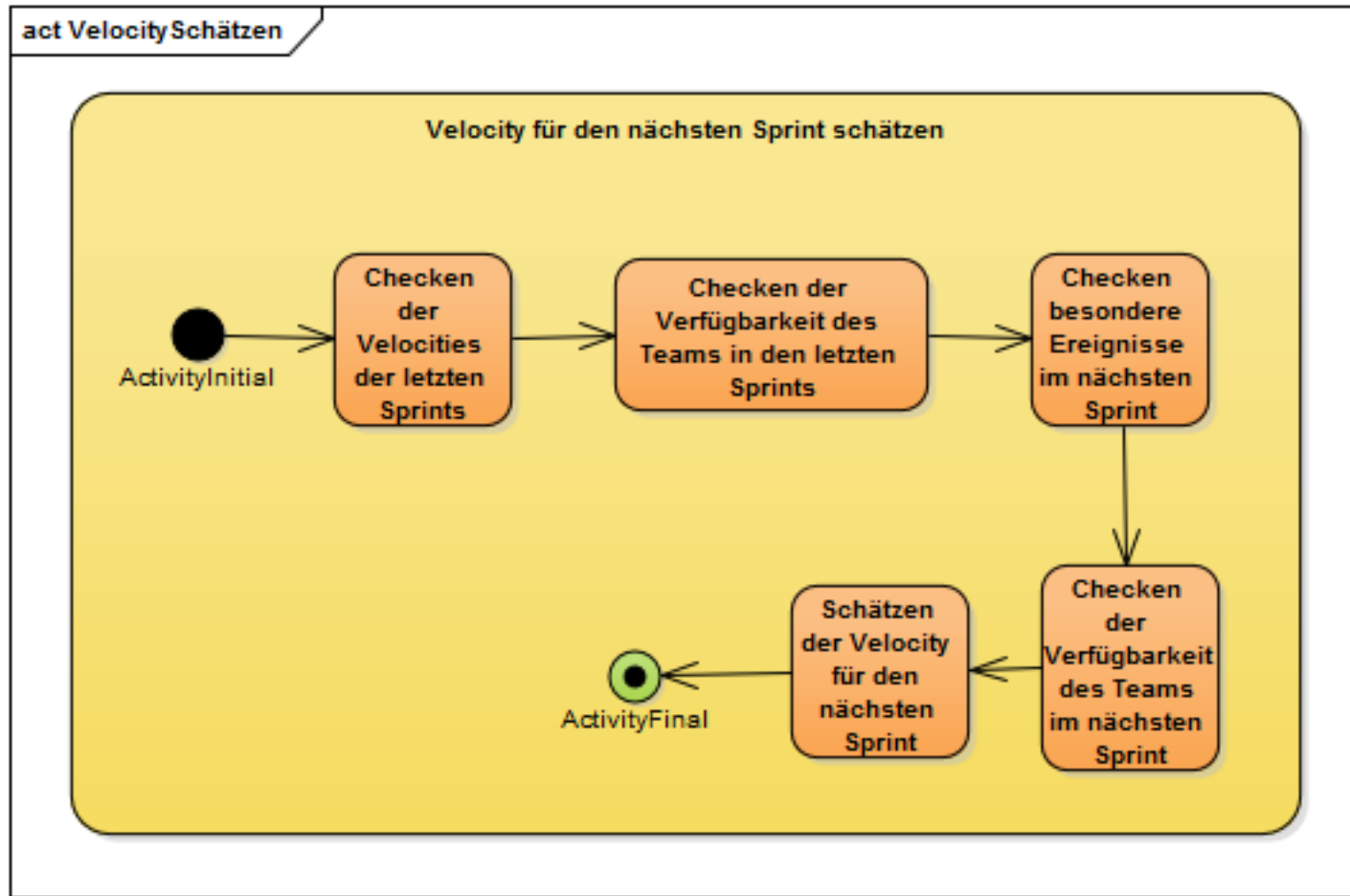
Sinnvolles Schätzen in Scrum



Planen in Scrum



Velocity schätzen



Scrum – die Artefakte

- **Product Backlog**

Das Product Backlog ist eine geordnete Liste von allem, was in dem Produkt enthalten sein kann.

- **Sprint Backlog**

Das Sprint Backlog ist die Menge der für den Sprint ausgewählten Product Backlog-Einträge.

- **Sprint Ziel:** Die ausgewählten Product Backlog-Einträge bilden eine zusammenhängende Funktionalität, die als Sprint-Ziel angesehen werden kann

- **Produkt Inkrement:** Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Product Backlog-Einträgen

Weitere Artefakte – Nach Gloger:

- Vision, Tasks, Releaseplan, Impediment Backlog

Definition of Done

Es müssen alle verstehen, was „Done“ bedeutet, sobald ein Product Backlog-Eintrag oder ein Produkt-Inkrement als „Done“ bezeichnet wird. Alle Teammitglieder müssen ein gemeinsames Verständnis davon haben wann Arbeit fertig ist, um Transparenz zu gewährleisten.

Dies erfolgt durch die **Definition of Done** des Scrum Teams.

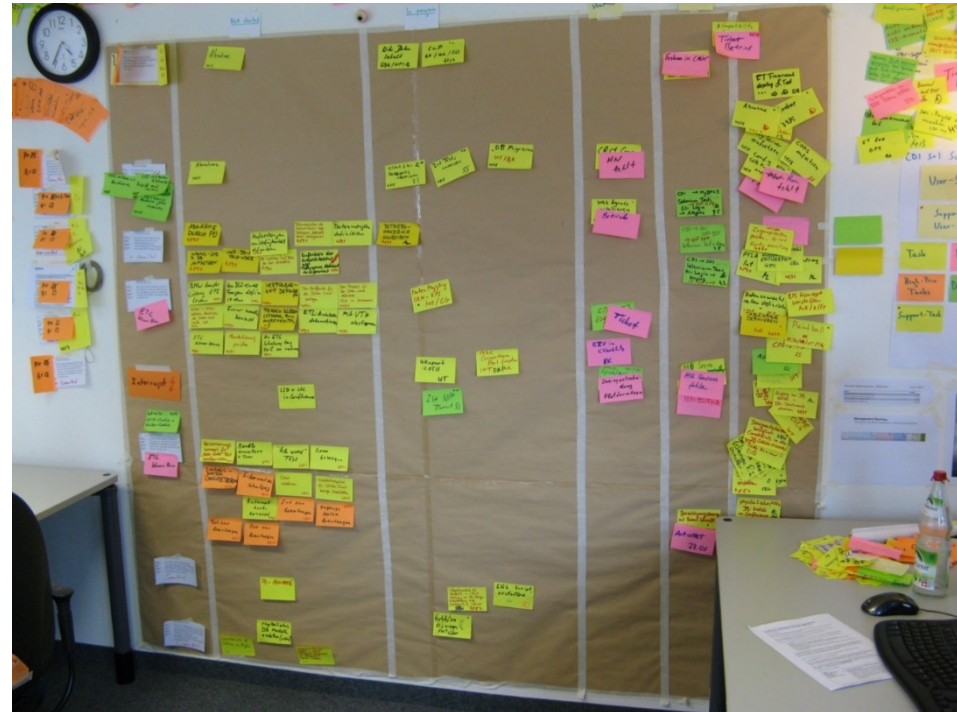
Scrum - Reporting

Der Projektfortschritt und die Prognosen beruhen auf verschiedenen Berichten:

1. Taskboard (= Scrum Board)
2. Verschiedene Formen von Charts
3. Berechnung der Velocity

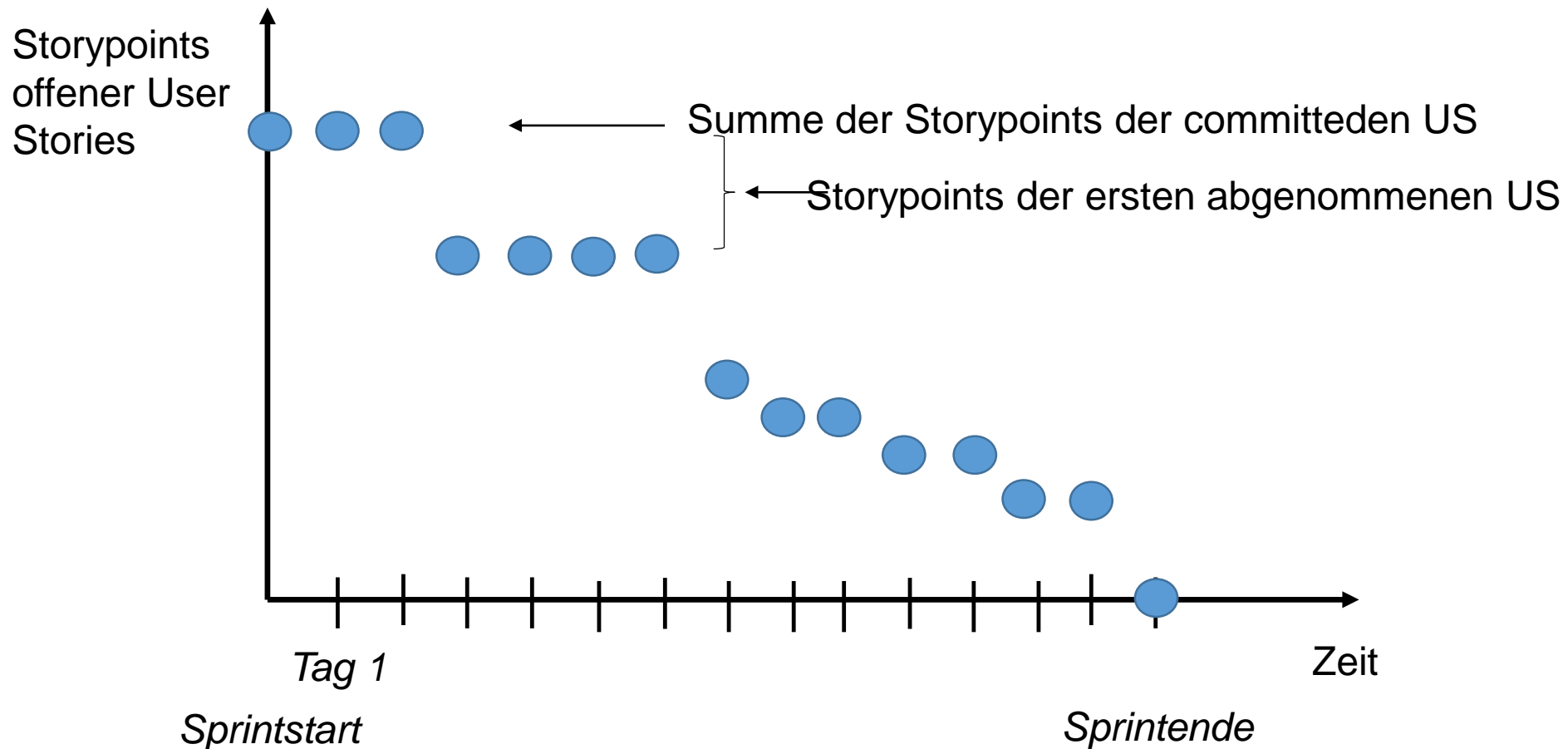
Task Board

- Täglicher Fortschritt
- Impediments sichtbar

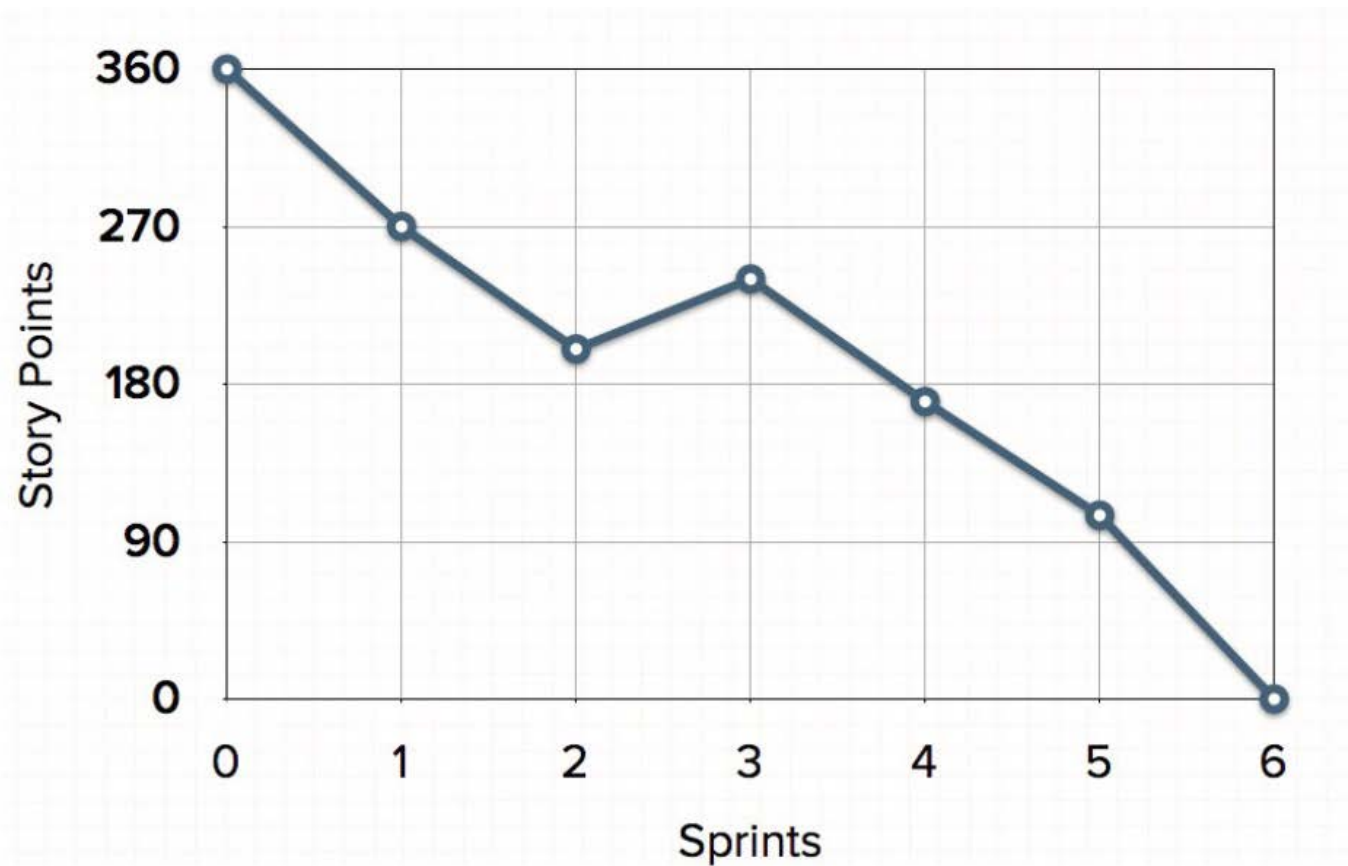


Fortschrittsverfolgung im Sprint

Sprint Burndown Chart - schematisch



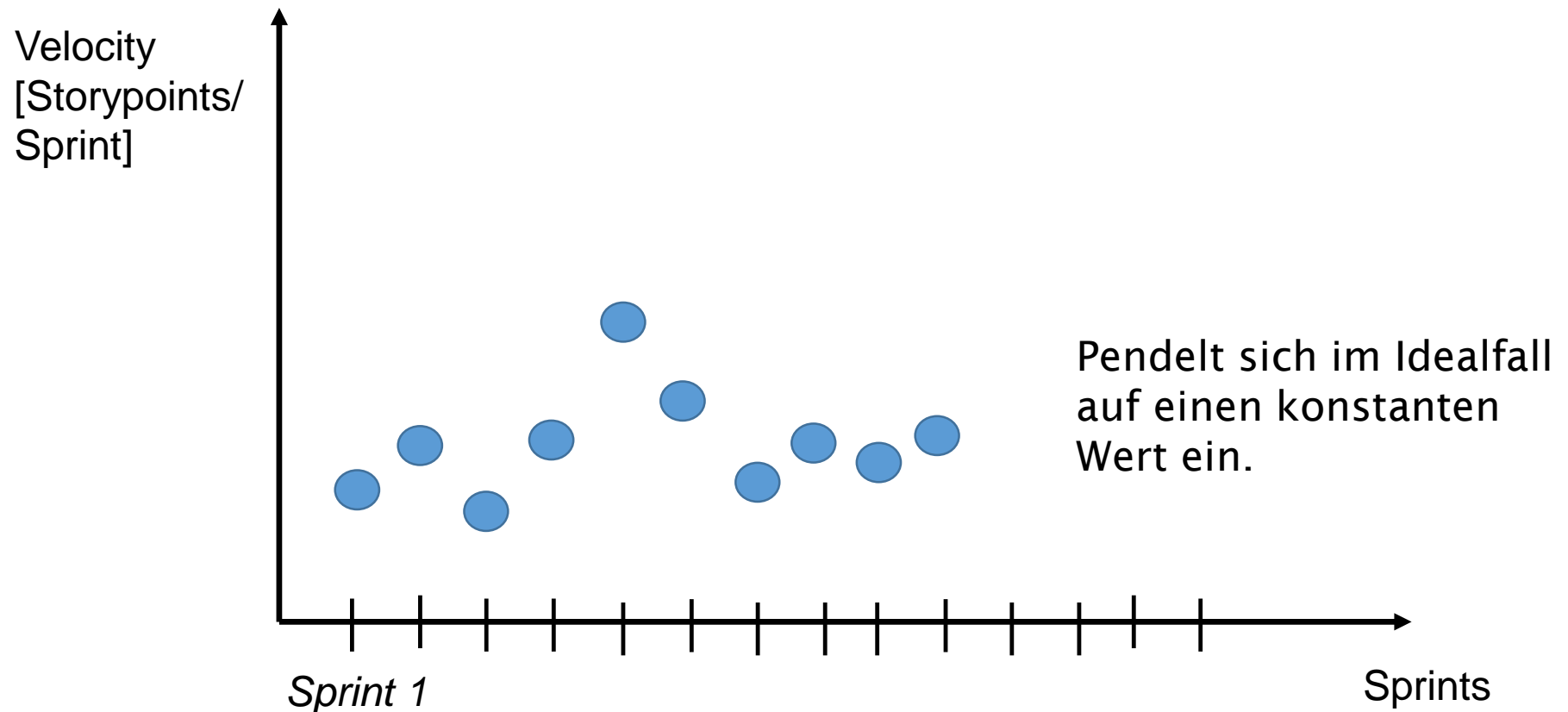
Release Burndown Chart



Aus <http://www.mountaingoatsoftware.com/agile/scrum/release-burndown>

Velocity

Velocity Chart - schematisch



- „Commitment“
- Methoden:
 - Continuous Integration → Siehe Kapitel Konfiguration Management
 - Hohe Testautomatisierung

Gliederung

- [Einführung in Scrum](#)
- [Einschätzung](#)
- [Referenzen](#)

- Agiles Vorgehen bedeutet nicht auf Planung zu verzichten.
- Scrum ist gut geeignet um Transparenz zu schaffen.
- Kein Allheilmittel.
- Es ist ein Fehler, aus einem Vorgehensmodell eine Religion zu machen.

Gliederung

- [Motivation für Scrum](#)
- [Einführung in Scrum](#)
- [Einschätzung](#)
- [Referenzen](#)

Referenzen

- The Home of scrum: <https://www.scrum.org/>
- Scrum Guide by Ken Schwaber and Jeff Sutherland:
<https://www.scrum.org/Scrum-Guide>
- <http://de.wikipedia.org/wiki/Scrum>
- R. Wirdemann: *Scrum mit User Stories*, 3. Auflage, Hanser, 2017
- B. Gloger; *Scrum*, Hanser, 2011
- Forrester: *Integrated Thinking: The Answer To IT's Perpetual Struggle*, Oct. 2013:
http://www.effectiveui.com/downloads/publications/EffectiveUI_Study_Integrated_Thinking.pdf

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

Literatur:

- Larman: Practices for Scaling in Lean & Agile Development, Addison-Wesley Professional
- B. Gloger: Scrum, 3. Auflage, Hanser
- Material unter http://www.craiglarman.com/wiki/index.php?title=Main_Page

Große Projekte

Was ist mit großen Projekten und Scrum?

➔ Mehr als ein Scrum Team benötigt.

Vor dem kommenden die Warnung :

Wenn es sich vermeiden lässt, dann lassen Sie das bleiben!

Aufbau von mehreren Scrum Teams

Nach Gloger prinzipiell zwei Arten:

- Organisches Wachstum

- Einarbeitung neuer Mitglieder, Team entscheidet, wann es sich teilt.

- Sprunghaftes Skalieren

- Die Mitglieder des initialen Teams übernehmen die Rolle des (Sub) Produkt Owners in den neuen Teams.

Wie werden die Teams geschnitten?

Nach welchen Kriterien werden die Teams geschnitten?

Prinzipiell zwei Möglichkeiten:

- **Component Teams**
 - Verantwortung für technische Komponenten
- **Feature Teams**
 - Verantwortung für fachliche Funktionen

Synchronisation durch:

- Scrum of Scrums
- Product Owner Team
- Scrum Master Group
- Virtuelle Teams für spezifische querschnittliche Aspekte (Architektur, Dokumentation, Tests, ...)
- Gemeinsame Planning Meetings
- Evtl weitere Meetings

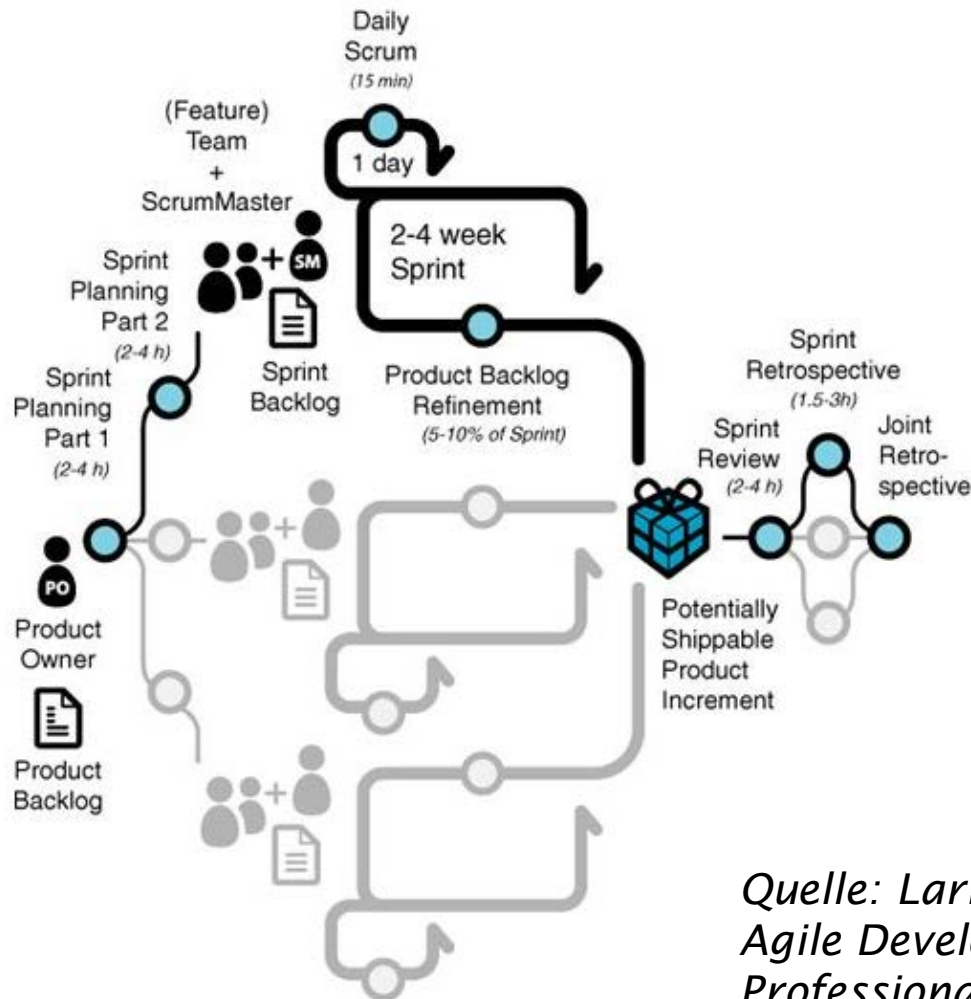
Wenn mehrere Scrum Teams existieren, wie ist die Organisation, welche Rollen existieren für die Teams?

Nach *Larman: Practices for Scaling in Lean & Agile Development*, Pearson

Vorschlag: Probier's damit:

- FW 1 für bis zu 10 Teams → siehe folgende Folie
- FW 2 für größere Teams → siehe übernächste Folie

FW1 für bis zu 10 Teams



- Ein gemeinsamer PO
- (Feature) Teams
- Scrum Masters für jedes Team

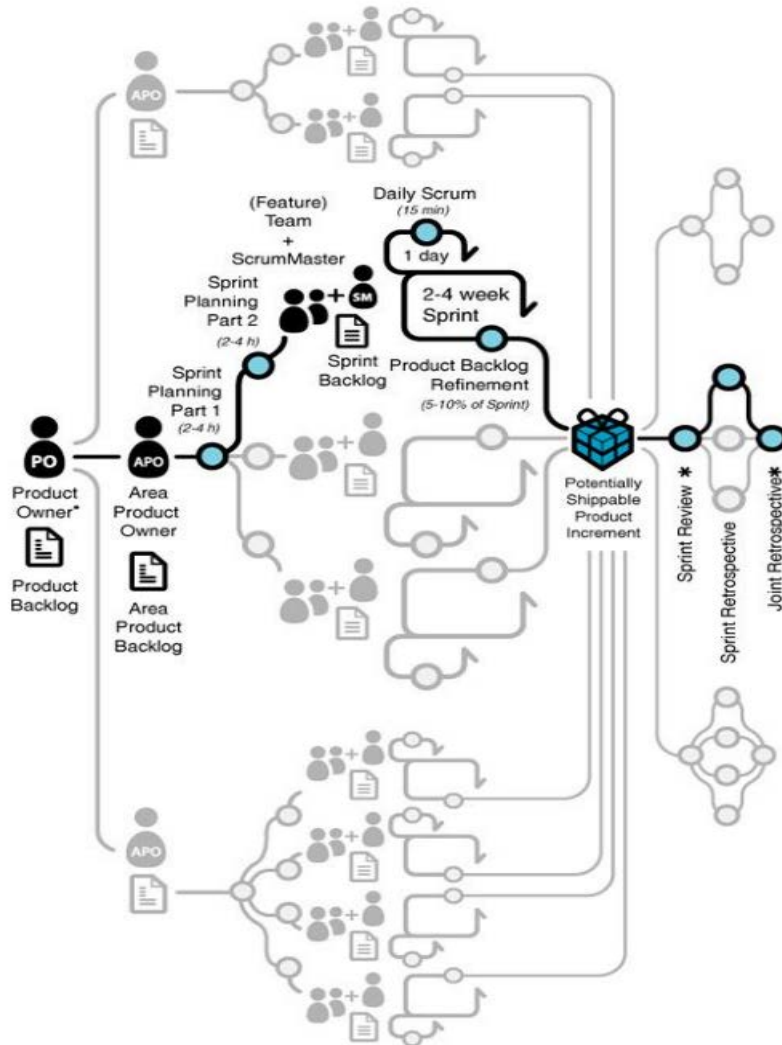
- 1 Product Backlog
- Sprint Backlogs pro Team
- 1 Product Increment

- Sprint Planning
- Daily Scrum
- Product Backlog Refinement
- Sprint Review
- Sprint Retrospectives
- Joint Retrospective

- Eine gemeinsame DoD

Quelle: Larman: Practices for Scaling in Lean & Agile Development, Addison-Wesley Professional

FW 2 für viele Teams



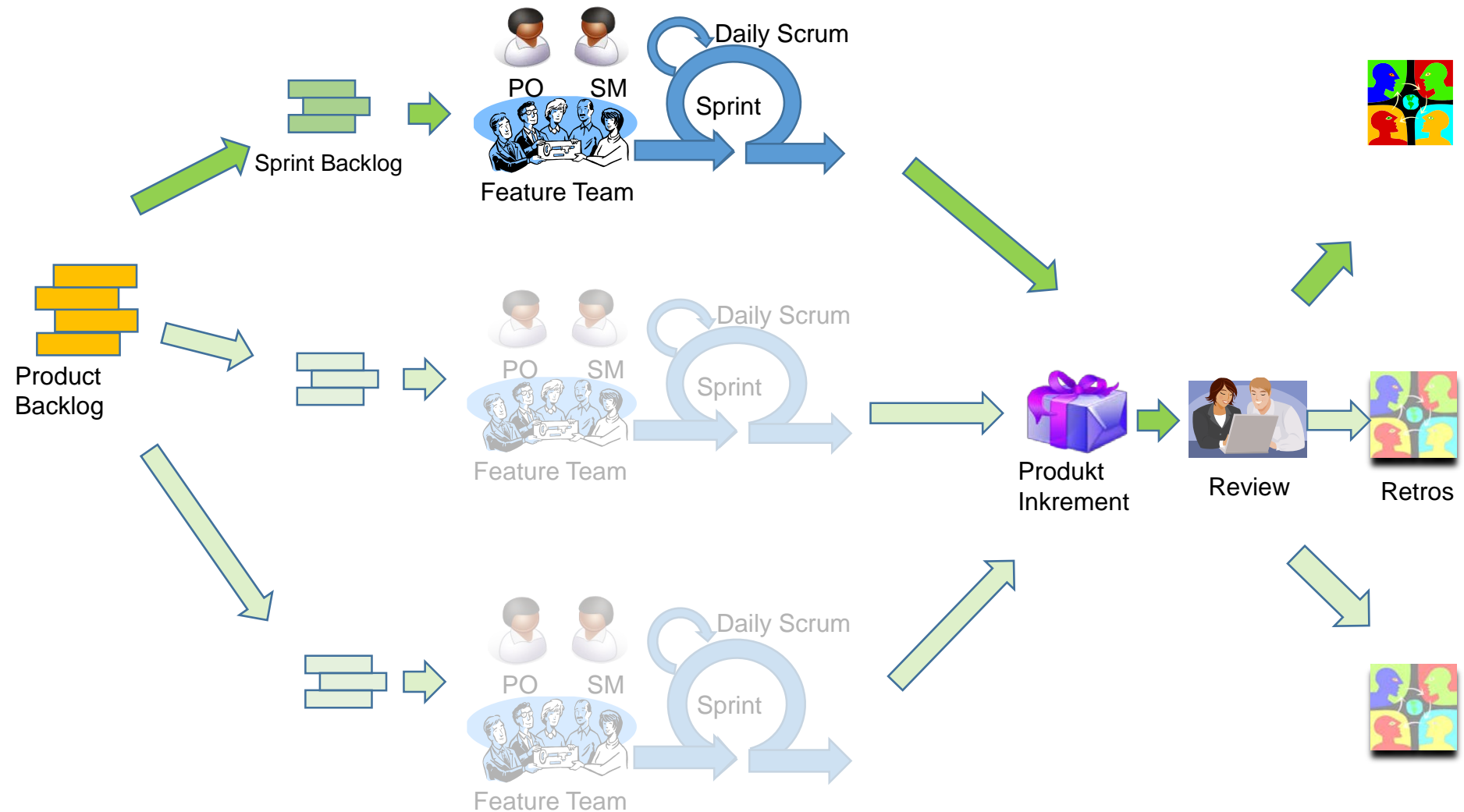
- Baut auf FW1 auf.
 - Im Prinzip ein Satz an FW1 Gruppen
-
- Aufteilung der Teams in requirement areas
 - PO plus Area Product owners
 - Gruppen von FW 1 Organisationen

Quelle: Larman: Practices for Scaling in Lean & Agile Development, Addison-Wesley Professional

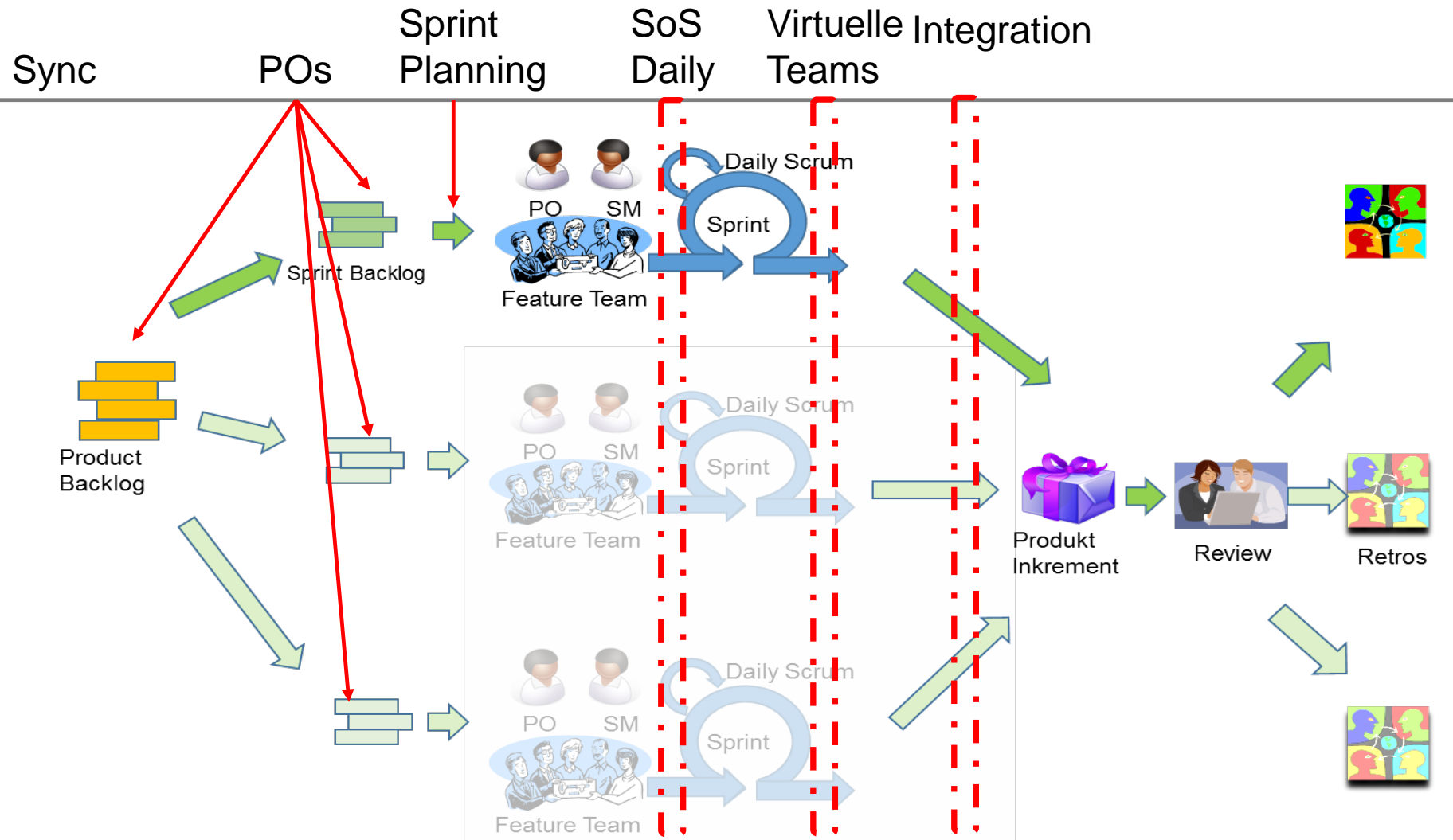
Beispielimplementierung von Scrum in großen Projekten

- Projekt mit bis zu 17 Scrum Teams
- Insgesamt bis zu 200 Projektbeteiligte
- Projekt unternehmenskritisch
- Mehrere Fremdfirmen beauftragt.

Scrum in großen Projekten - Bsp



Scrum in großen Projekten



Tägliche Synchronisation durch SoS



- **Projektleitung**
 - Trotz Scrum Projekt übergeordnete Projektleitung
- **Tests**
 - Im Bsp Projekt: Eigenes Testteam
- **Integration**
 - Im Bsp Projekt: Eigenes Team für Integration zu Sprint Ende
- **Entstehung der Teams:**
 - Durch die Projektleitung bestimmt

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
- Agiler Festpreis

Vertragsmodelle

Die Wahl eines Preismodells hängt von verschiedenen Faktoren ab:

- Projektart (Neuentwicklung, Anpassung, Weiterentwicklung usw.)
- Sicherheitsbedürfnis und Planbarkeit (stabiler Preis, Haushaltsplanung)
- Konkurrenzfähigkeit (möglichst niedriger Preis)
- Gewinnchance (für den Auftragnehmer)
- Nutzenerwartung (Nutzen soll höher sein als Kosten)
- AG-AN- Vertrauensverhältnis

Prinzipielle Unterscheidung:

- Festpreis
- Aufwandspreis (Time&Material)

Varianten

- Aufwandspreis mit Obergrenze
- Phasenfestpreis
- Agiler Festpreis
- Festpreis mit inhaltlichem Spielraum

- Überblick (Wiederholung)
 - Einleitung
 - Wichtige Vorgehensmodelle – nicht agil
- Agile Methoden
 - Agile Methoden allgemein
 - Wiederholung: Scrum
 - Scrum in großen Projekten
- Vertragsmodelle
 - Agiler Festpreis

Der Agile Festpreis

Literatur:

- Opelt et al.: *Der Agile Festpreis*, Hanser, 2012

Extern vergebene Leistungen

1. Ausschreibung



Kunde



Dienst
leister

2. Angebote



3. Auftragsvergabe



Auftragnehmer vs. Auftraggeber

Auftraggeber:

- Will abschätzen, was er für sein Geld bekommt.
- Will wissen, was ein System kosten wird.
- Will den günstigsten Anbieter auswählen können.
- Muss dafür Budget freischalten.
- Will sein Risiko minimieren.

Auftragnehmer

- Will abschätzen, wieviel er verdienen kann.
- Will die Mitbewerber ausstechen.
- Will sein Risiko minimieren.

Auftraggeberinteressen:

- Tendenz zu Festpreis
 - → Risiko beim Auftragnehmer

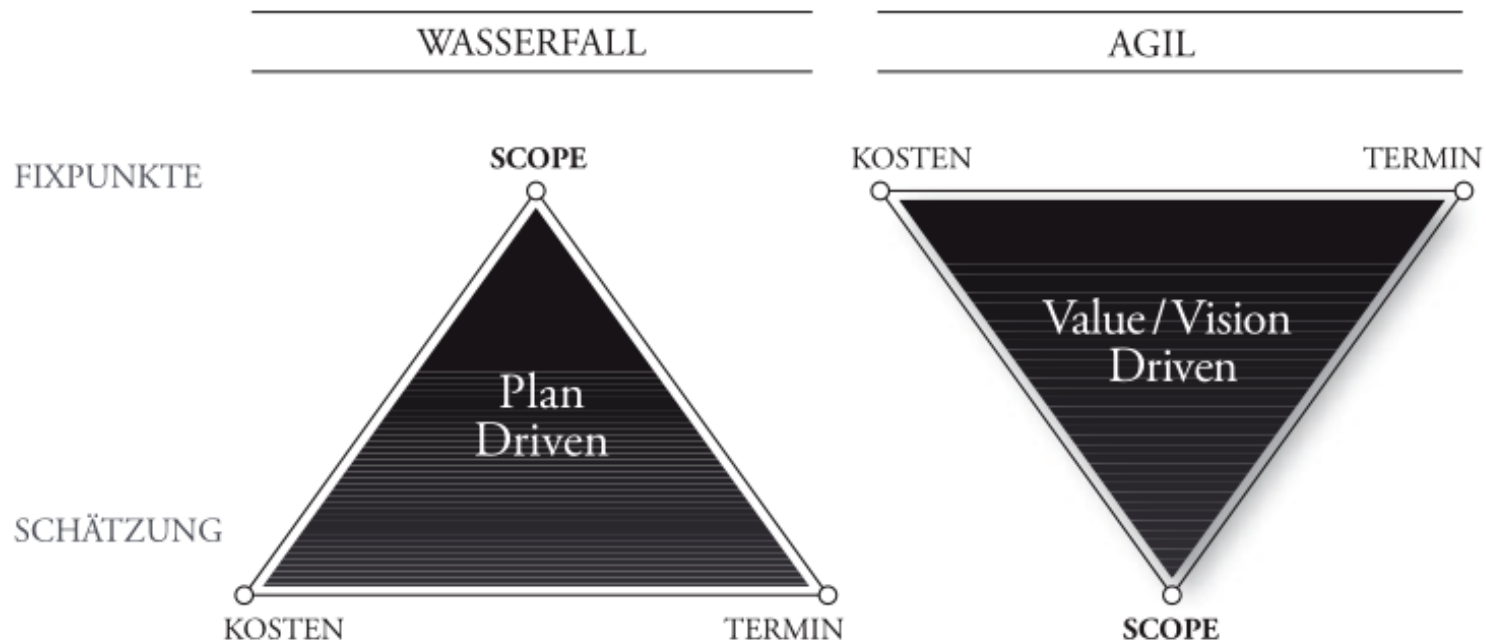
Auftragnehmerinteressen:

- Tendenz zu Aufwandsvertrag
 - → Risiko beim Auftraggeber

→ Lösungsversuch: Agiler Festpreis als neues Vertragsmodell.

Agiler Festpreis

Aus: „Der Agile Festpreis“



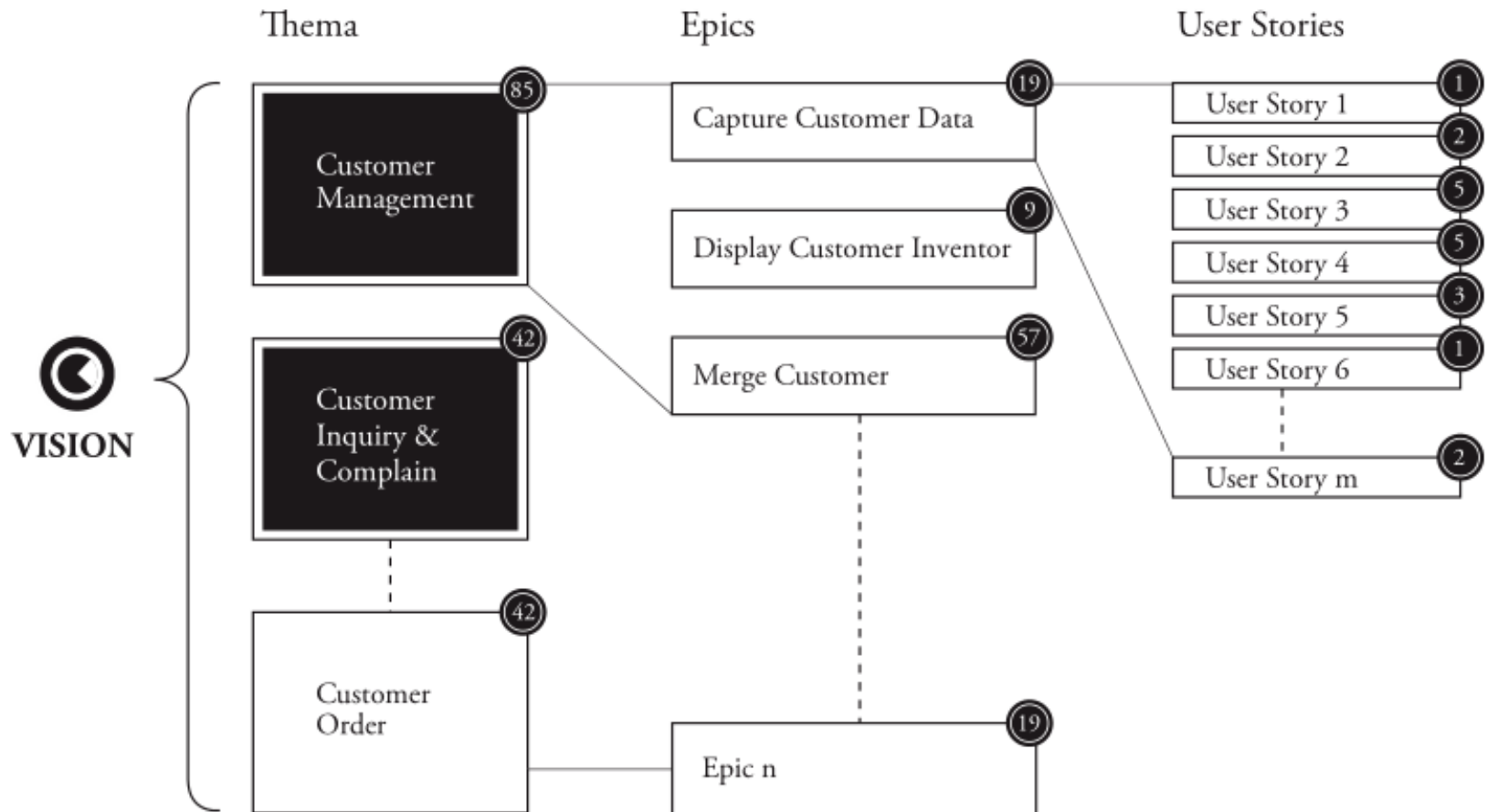
Wie kommt man zu einem agilen Festpreisvertrag?

Nach „Der Agile Festpreis“ kommt man mit folgenden Schritten zu einem Agilen Festpreisvertrag

1. Definition des Vertragsgegenstands auf grobgranularer Ebene.
(Aus Sicht des Anwenders, vollständig, aber nicht detailliert)
2. Detailspezifikation einer exemplarischen Menge an Referenz User Stories.
3. Gemeinsamer Workshop zum Gesamtscope → indikativer Festpreisrahmen
4. Definition von Riskshare, Checkpoint Phase und Ausstiegspunkten
5. Vereinbarung zur Scope Governance
6. Definition von Motivationsmodell und Kooperationsmodell

Quelle: Der agile Festpreis, Hanser 2012

Detallierung der Vision



Quelle: Opelt et al: Der agile Festpreis

Wie kommt man zu einem agilen Festpreisvertrag?

SCOPING



BILD 3.2 Scoping und Prozessdefinition für den Agilen Festpreisvertrag



PROZESSE

Aus „Der Agile Festpreis“, Seite 46