

# Projektmanagement

Ein kurzer Blick auf „Agile“

---

Teil 12

- Projektmanagement - WS 2015/16

Jörg Pechau

Department Informatik, Uni Hamburg

# Agenda

---

- Organisatorisches
- Kurze Erinnerung
  - Menschen im Projekt
  - Letzter Punkt
  - „Wrap Up“ der („weichen“) Erfolgsfaktoren
- Agile
- Lesestunde
- Übungsblatt

# Kurze Erinnerung

# Menschen im Projekt

---

- Phasenmodell nach Tuckmann



(„Weiche“) Erfolgsfaktoren

„Wrap Up“





# Agile Methoden

## Begriffsklärung

# Ursprünge

---

- Kaizen (改善), nach 1945, Japan: „Improvement“ (~ Verbesserung, Verfeinerung)
  - Ziel „Veränderung zum Besseren“
  - Durch kontinuierlichen Veränderungsprozess Verbesserung erzielen
  - Bekanntes Beispiel war Toyota
- XP (eXtreme Programming), ca. 1999, USA: Software Qualität erhöhen durch
  - Kundenfokus (On site Customer)
  - Änderungsorientiertes Vorgehen (Embrace Change)
  - Verbesserung der Software-Praxis, z.B.
    - Pair Programming, TDD (Test First, Unit Tests etc.), Continuous Integration, Collective Code Ownership, Refactorings, Small Releases, Planning Game, Sustainable Pace, Simple Design (YAGNI)
  - Bei Chrysler aus einem Projekt (C3) heraus entwickelt



# Manifesto for Agile

---

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

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

© 2001, the above authors  
this declaration may be freely copied in any form, but only in its entirety through this notice.

<http://agilemanifesto.org>



# Andere Agile Methoden (Auszug)

---

- Software Entwicklungsmethoden
  - XP - eXtreme Programming
  - BDD - Behaviour Driven Development

- TDD - Test Driven Development

- Lean Software Development

Firmen prägen über die Zeit meist einen eigenen Methoden-Kanon aus.

- Crystal

- (Projekt-) Managementmethoden
  - Kanban
  - SCRUM

# Manifesto for Software Craftmanship (2009)

---

## **Raising the bar**

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software, but also well-crafted software

Not only responding to change, but also steadily adding value

Not only individuals and interactions, but also a community of professionals

Not only customer collaboration, but also productive partnerships

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

© 2009, the undersigned.  
this statement may be freely copied in any form,  
but only in its entirety through this notice.

<http://manifesto.softwarecraftmanship.org/>

# Bewertung agiler Methoden

---

- Wertesysteme „Agile Manifesto“ und „Manifesto for Software Craftmanship“ erinnern an einen „hypokratischen Eid“ für Software-Entwickler
- Nützt bei vielen Projekttypen und schadet zumindest nicht

- Hoher Wert bei Projekten mit hoher Änderungswahrscheinlichkeit, z.B. explorative Projekte

- Projekte mit kundenspezifischen Scope gewinnen durch „Agile“ häufig mehr oder nicht viel

Agiles Vorgehen entpuppt sich häufig als Motivations-Booster bei den Teams!

- Häufiges Feedback hilft

- Optimalen Kundennutzen zu erreichen

- Risiko durch Fehlentwicklung zu reduzieren

- Probleme in Organisation und Aufgabe frühzeitig aufzuzeigen





# Scrum

Ein Agiles (Projekt-)  
Management Rahmenwerk



# Was ist Scrum?

---

- Scrum
  - Ist ein (Projekt-) Managementrahmenwerk
  - Komplettiert und komplementiert Software-Entwicklungs-Methoden um Projektmanagementanteile
  - Hat wird mittlerweile häufig als Synonym agilen Vorgehens wahrgenommen
  - Ist „Mainstream“

# Scrum ist empirisches Management

---

- Kernideen

- „Inspect and adapt“

- Selbstorganisation

- Selbstmanagement

Erfordert Transparenz, den Mut loszulassen  
und Änderungswillen!

- Eigenverantwortung

- Autonome Teams

- Fokus auf Erreichung des optimalen Nutzwerts für Kunden



Rollen

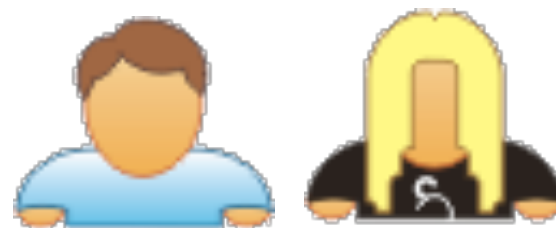
Projektorganisation

# Das Scrum Team

Product Owner



Scrum Master



Umsetzungs Team



# Projektorganisation

---

Kunden



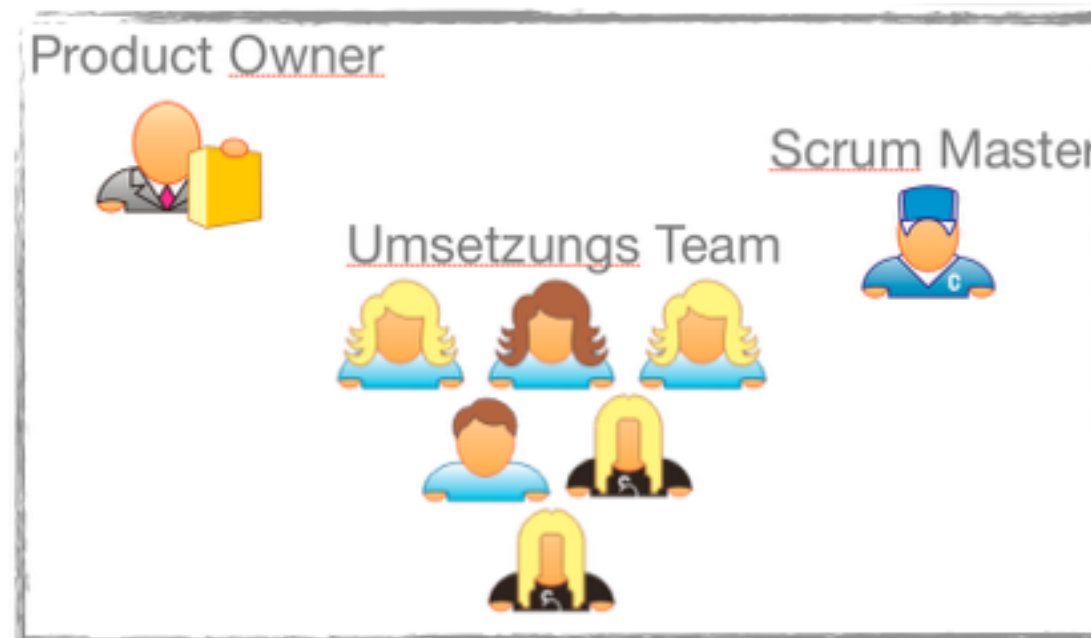
Agile Coach



Management

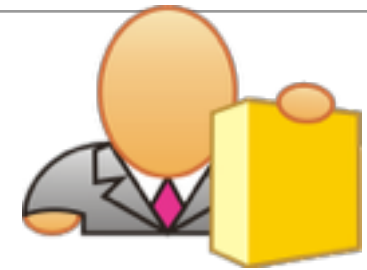


Scrum Team



# Der Product Owner

---



- Repräsentiert den Kunden, bzw. ist der Kunde
- Definiert das Produkt
  - Beschreibt den Funktionsumfang in Form von Epen, Themen, User Storys
  - Sammelt diese im Product Backlog
  - Pflegt und priorisiert das Product Backlog
- **Der Product Owner trägt die inhaltliche und kommerzielle Verantwortung!**
  - **Er ist kein Projektleiter!**
  - Priorisiert Features, z.B. abhängig vom Markt
  - Passt Features und Prioritäten nach Bedarf für jeden Sprint an
- Ist verantwortlich für das finanzielle Ergebnis des Produkts (ROI)
- Akzeptiert Sprint Ergebnisse entsprechend Sprint Ziele und DoD
  - Definiert zusammen mit Team und Scrum Master die Definition of Done (DoD)
  - Definiert zusammen mit Team die Sprint Ziele für jeden Sprint

# Der Scrum Master

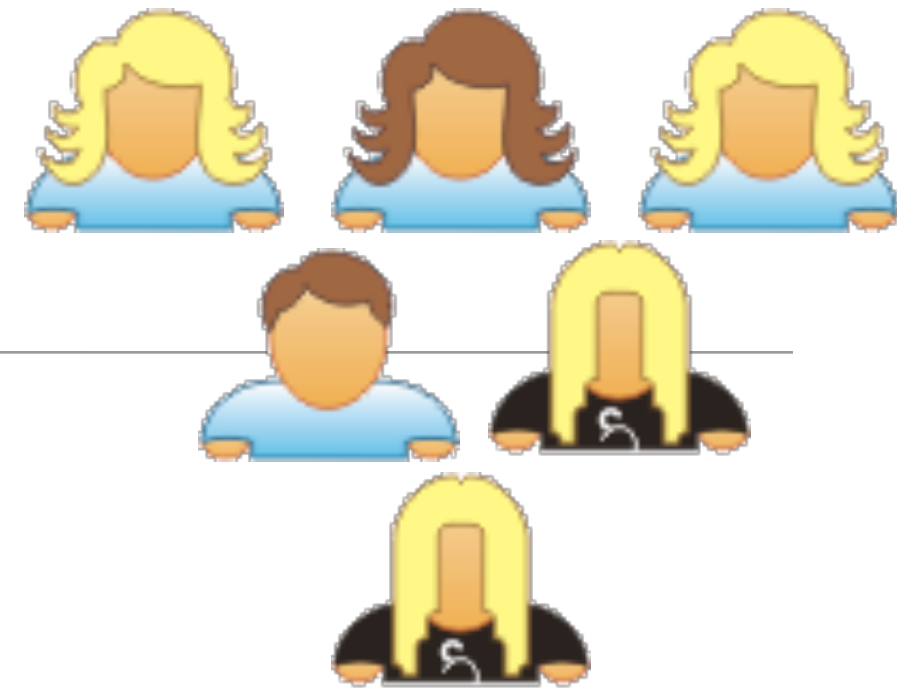
---



- Repräsentiert das Management gegenüber dem Projekt
  - Unterstützt Team und Product Owner, aber managed nicht
    - Ist Team Coach
  - Schützt das Team vor äußeren Störungen und beseitigt Hindernisse
  - Verantwortlich für die Einhaltung von Scrum-Werten und -Techniken
  - Stellt sicher, dass das Team vollständig, funktional und produktiv ist
  - Unterstützt die enge Zusammenarbeit zwischen allen Rollen und Funktionen
  - Kümmerst sich um Hindernisse (Impediments / Impediment List)
- Der Scrum Master verantwortet den Prozess ist aber kein Projektleiter!
- Verbessert die engineering practices

# Das Team

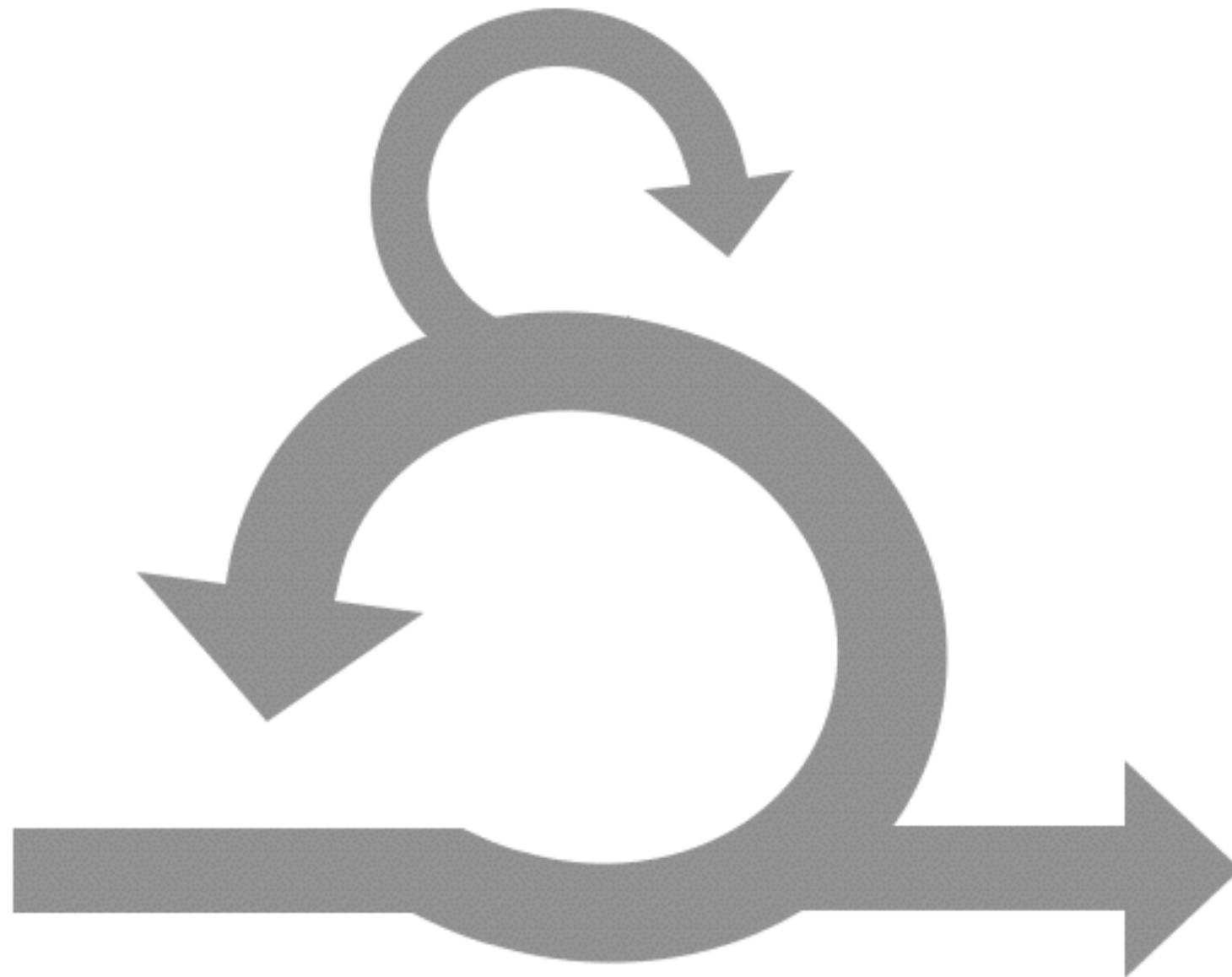
---



- Setzt das Projekt um.
  - Das Team arbeitet selbstorganisiert und managed sich selbst
    - Beurteilt wie viel Arbeit pro Sprint leistbar ist
    - Wählt Einheiten aus dem priorisierten Product Backlog aus
    - Schätzt die Einheiten
    - „Committed“ sich
- Das Team verantwortet die Umsetzung

- Teamaufbau
  - Größe: 3 - 5 - 7 (maximal 9, idealerweise nicht gerade)
  - Funktionsübergreifend, autonom
  - Vollzeitmitglieder, bis auf Ausnahmen, z.B. Experten deren Hilfe nur kurz nötig ist
  - Zusammensetzung ändert sich - wenn überhaupt - nur nach einem Sprint nicht während





Die Artefakte

Womit wir arbeiten

# Product Backlog

---

- Priorisierte Liste von „Product Backlog Items“
  - Sammlung der funktionaler und nicht funktionaler Anforderungen
  - Unterschiedliche Granularität
- Form:

Viele können (evtl.) das Product Backlog ergänzen,  
nur der Product Owner priorisiert es!

  - Epics, Topics, User Stories
- In der Sprint Planung „wandern“ Product Backlog Items in das Sprint Backlog
- Realisierungsaufwand wird in relativem, abstrakten Maß geschätzt

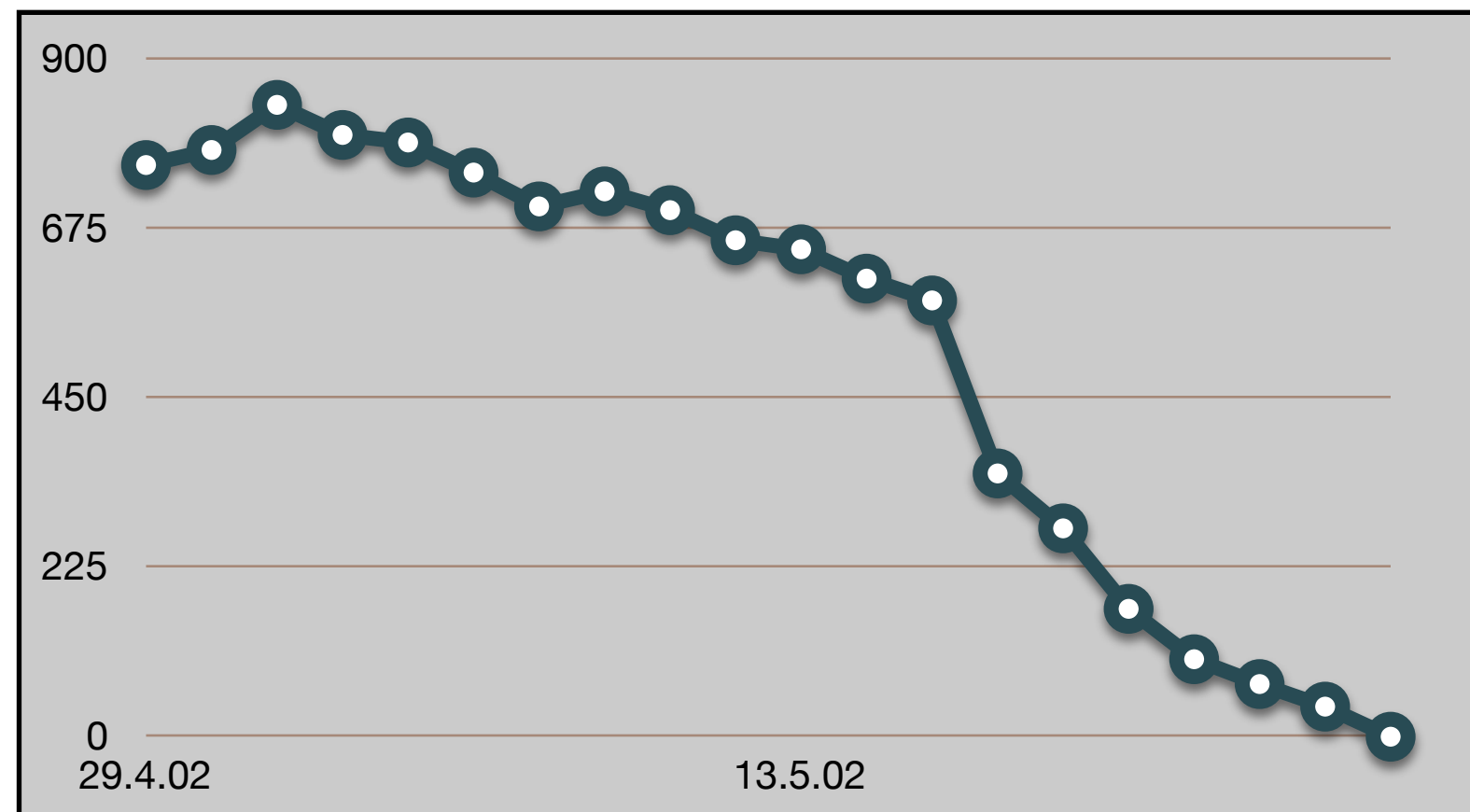
# Sprint Backlog

---

- Definiert die Aufgaben für einen Sprint
  - Um das Sprint Ziel zu erreichen und
  - Auf die sich das Team committed hat
- User Storys werden dazu in notwendige Tasks heruntergebrochen
- Tasks werden in Stunden geschätzt
- Während eines Sprints wird der Scope von außen nicht mehr geändert

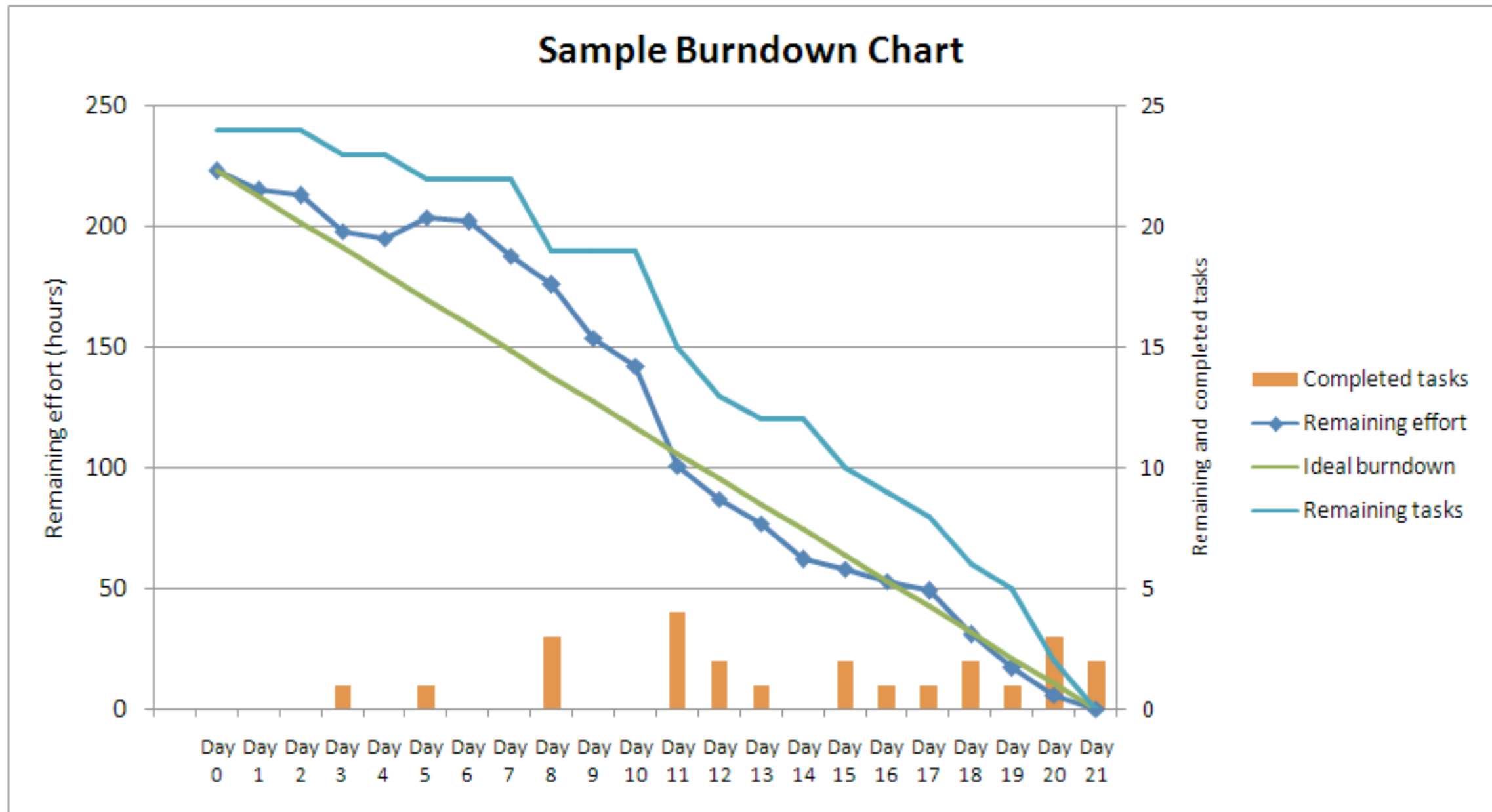
# Burndown-Charts

- Sprint Burndown
  - Y-Achse: Verbleibender Aufwand in Stunden
  - X-Achse: Fortschritt in Tagen
- Release Burndown, z.B.
  - Y-Achse: Verbleibende Aufgaben in Anzahl Product Backlog Items
  - X-Achse: Fortschritt in Sprints





# Yet another Burndown Chart



# Weitere Artefakte

---

- Scrum Board
- Release Plan
- Impediment List
- Definition of Done
- Ggf. Definition of Ready





# Der Prozess

Wie läuft ein Scrum Projekt ab?



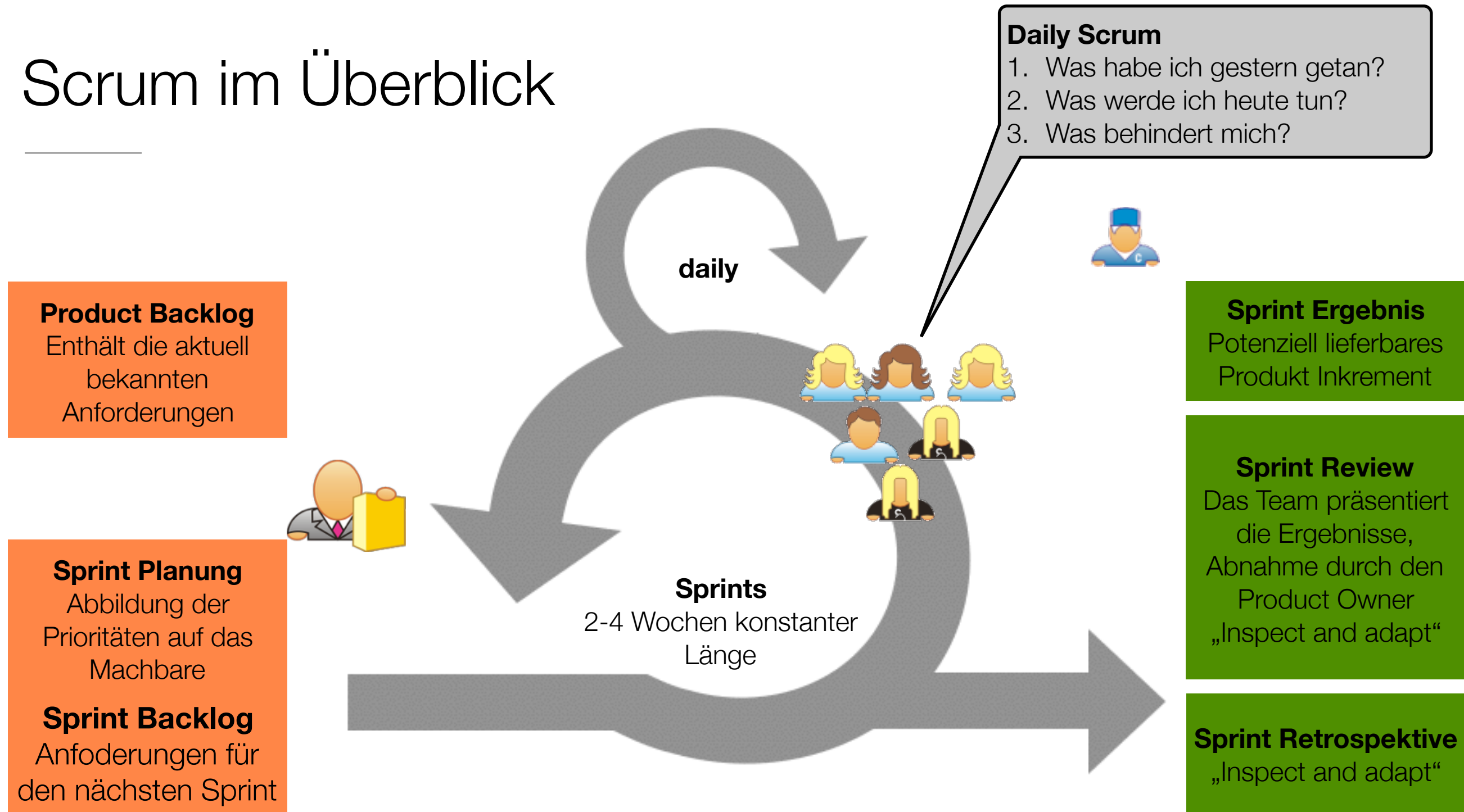
# Generelles Vorgehen

---

- Ziel:
  - Das „**minimal viable Product**“ (oder auch „minimal desirable Product“)
- Vorgehen
  - Time Boxing
  - Projekt wird unterteilt in Anzahl von Iterationen, Sprint genannt
    - Sprint sollten zwischen 1 und 3 Wochen dauern
    - Der Scope innerhalb eines Sprints ist fix, der Scope richtet sich nach der Team-Geschwindigkeit (Velocity)
    - Ein Sprint kann ggf. abgebrochen werden
    - Am Ende eines Sprints steht ein potentiell lieferbares Produkt
- Die „Meilensteine“ ergeben sich aus der sogenannten Release-Planung, die Ziele aber keine Detailplanung setzt



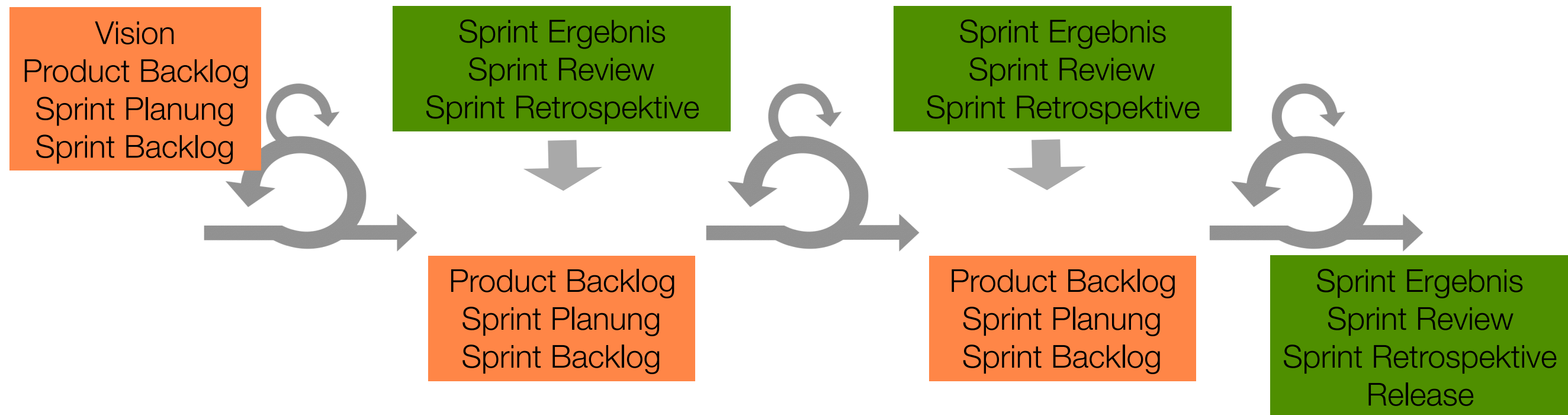
# Scrum im Überblick



## Sprint Ausführung

Das Team setzt selbstorganisiert und ungestört das Sprint Backlog um.  
Der Scrum Master hält Störungen fern, entfernt Hindernisse und achtet auf den Prozess.  
Der Product Owner steht für Fragen und Abstimmungen zur Verfügung.

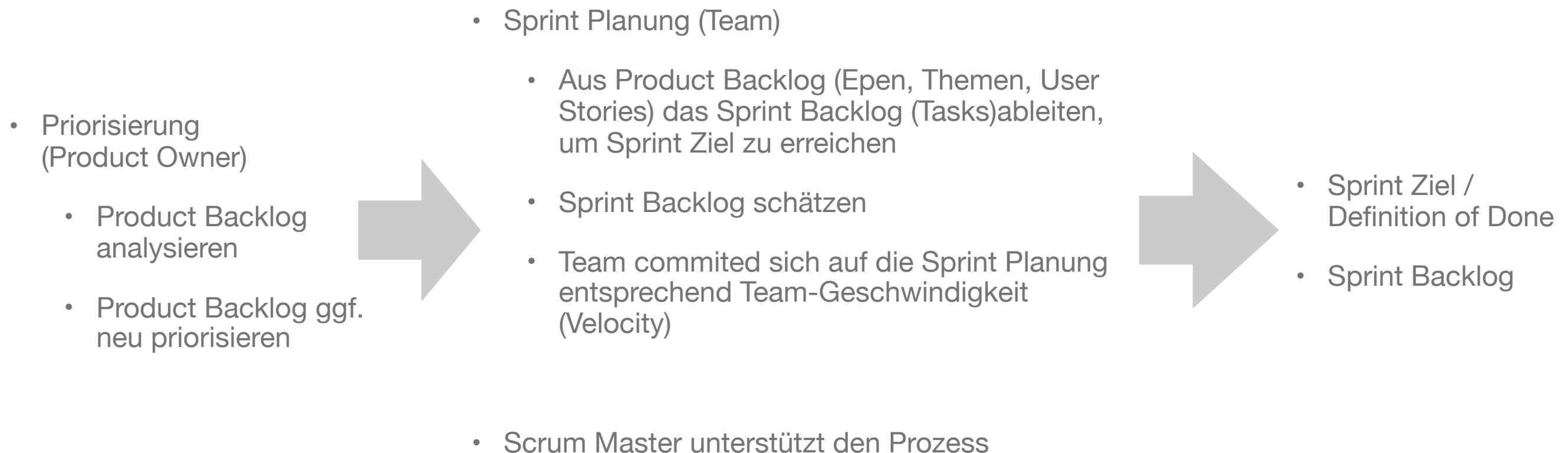
# Das Projekt als Folge von Sprints



- Scrum-Projekte schreiten in Serien von Sprints voran
- Typische Sprintdauer beträgt 2–4 Wochen
- Um sinnvolle Messdaten zu erhalten sollte Sprintdauer konstant bleiben
- Jeder Sprint liefert jeweils ein potentiell lieferbares Inkrement des Produkts
- Zwischen Sprints wird neu priorisiert, während der Sprints ist Scope fix
- Product Backlog wird kontinuierlich fortgeschrieben

# Sprint Planungsmeeting

---



# Daily Scrum

---

- Täglich, 15 Minuten, stehend
- Probleme werden ausserhalb der Runde gelöst, alle beantworten die Fragen
  - Was habe ich gestern getan?
  - Was werde ich heute tun?
  - Was behindert mich?
- Das ist kein Report, sondern Commitment dem Team gegenüber
- Offen für alle, aber nur das Scrum Team hat Rederecht
- Scrum Master sammelt Impediments auf

# Sprint Review

---

- Das Team präsentiert das Sprint Ergebnis am Ende des Sprints und erhält Feedback
- Teilnehmer: Scrum Team, Kunden, „Welt“...
- Dauer ca. 2h
- „Maximal 2h Vorbereitung“-Daumenregel
- Stets das Produkt zeigen keine „Slide-Work“, „Report“ etc.!
- Product Owner beurteilt das Ergebnis anhand
  - Des Sprint-Ziels und
  - Der Definition of Done
- Scrum Master organisiert und stellt den Prozess sicher

# Sprint Retrospektive

---

- Das Scrum Team prüft am Ende des Sprints, was gut und was nicht gut funktioniert
- Dauer: ca. 15-30 Min
- Teilnehmer: Team, Scrum Master
- Das Scrum Team diskutiert, wie es **Inspect & Adapt** möchte, z.B.
  - „Beginnen mit...“
  - „Aufhören mit...“
  - „Weitermachen mit...“



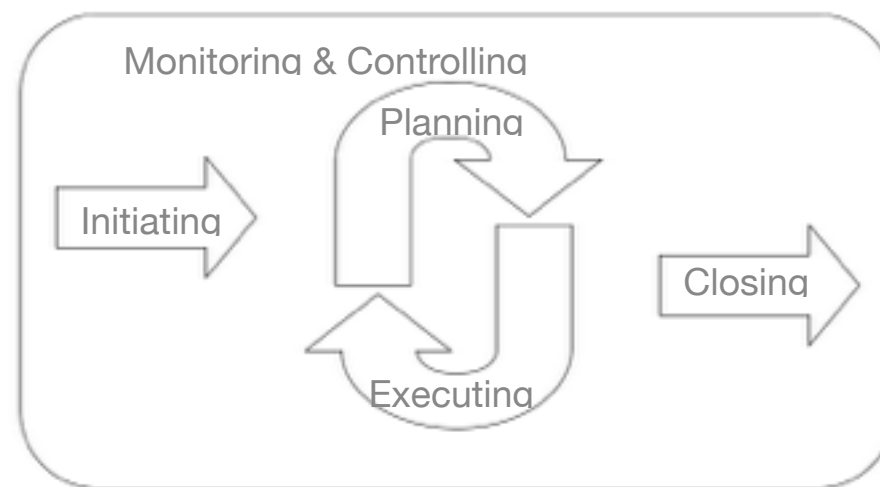
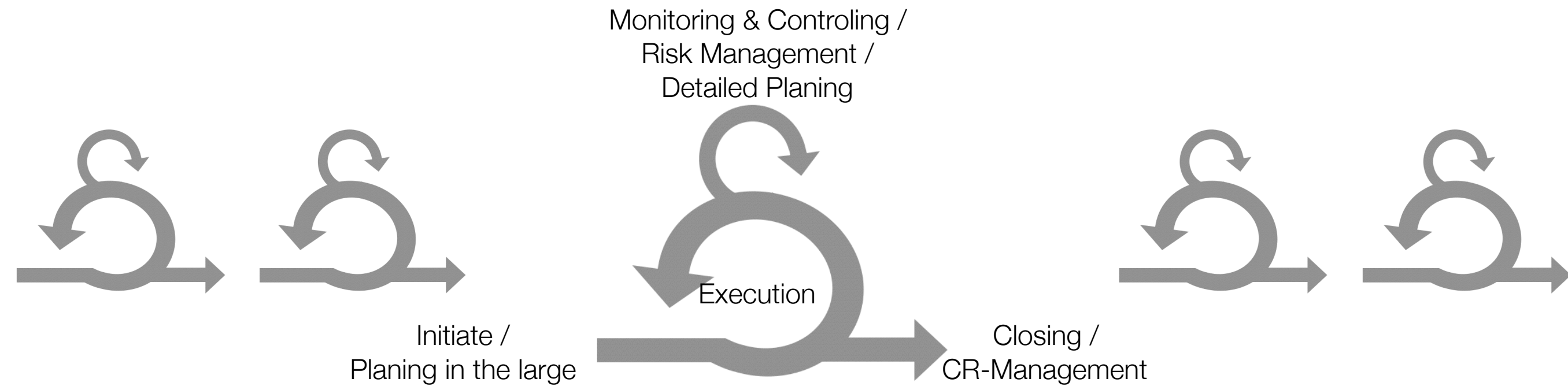


Agiles und klassisches  
phasenorientiertes PM

Eine grobe „Abbildung“

# Vorgehen stellt Kunden und Änderungen in den Mittelpunkt!

## Ungefährliches Mapping Prozessgruppen PM auf Scrum Prozess



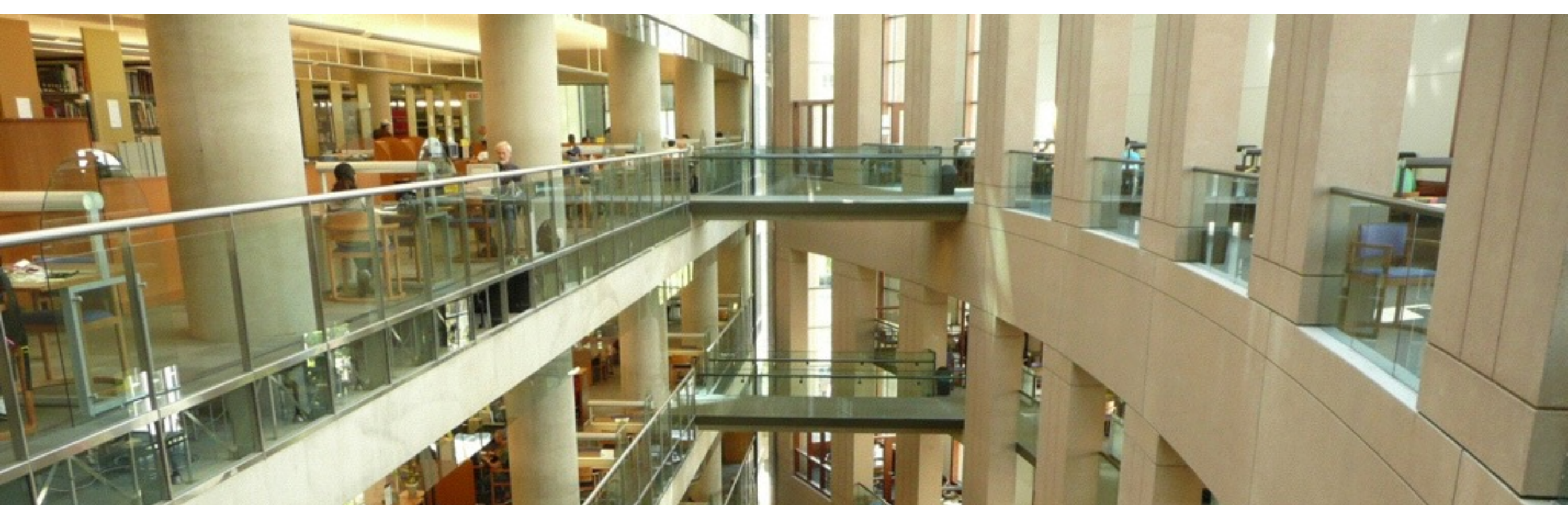
- Scrum-Projekte schreiten in Serien von Sprints voran
- Typische Sprintdauer beträgt 2–4 Wochen
- Um sinnvolle Messdaten zu erhalten sollte Sprintdauer konstant bleiben
- Jeder Sprint liefert jeweils ein potentiell lieferbares Inkrement des Produkts
- Zwischen Sprints wird neu priorisiert, während der Sprints ist Scope fix
- Product Backlog wird kontinuierlich fortgeschrieben

# Was macht Scrum so anders?

---

- Das Scrum-Vokabular, Rollen, Artefakte und Prozesse sind in ~„10 Min“ erlernt
  - Mächtig, einfach, aber nicht simpel
  - Wirkt einfach, erfordert aber viel Disziplin
- Der eigentliche Unterschied zu klassisch phasenorientiertem Projektmanagement liegt im zugrundeliegenden Wertesystem!
  - Gut bei explorativen Projekten, bzw. hoher Change-Wahrscheinlichkeit
  - Erfordert i.a. Haltungsänderung - das liegt nicht jedem...
- Agiles und nicht agiles Vorgehen lassen sich prinzipiell zu einem **hybriden Vorgehen** kombinieren, erfordert aber eine Reihe von Kompromissen



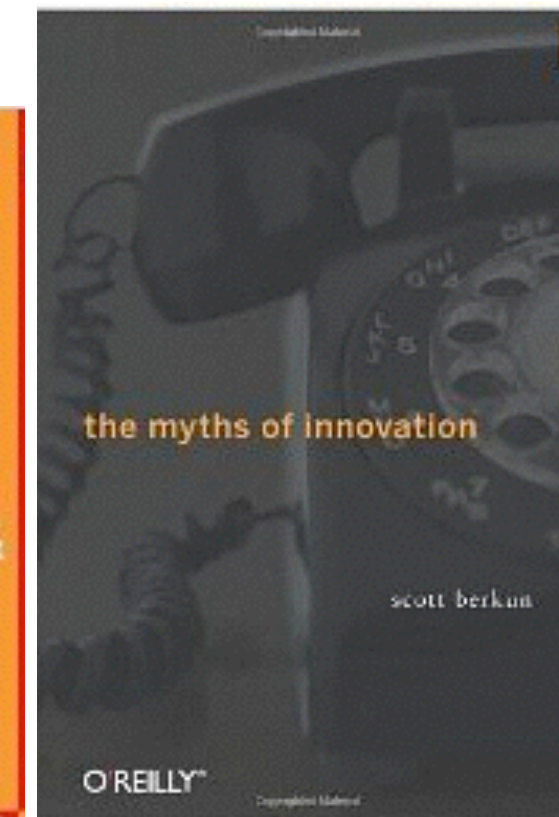
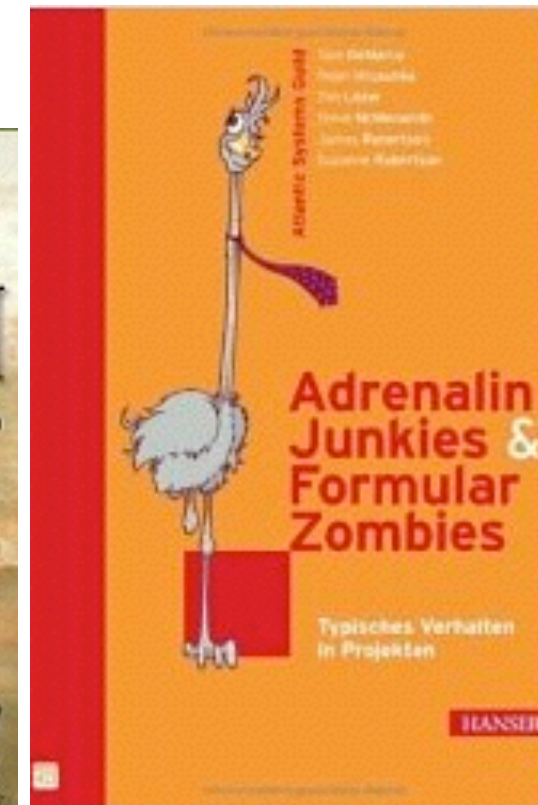
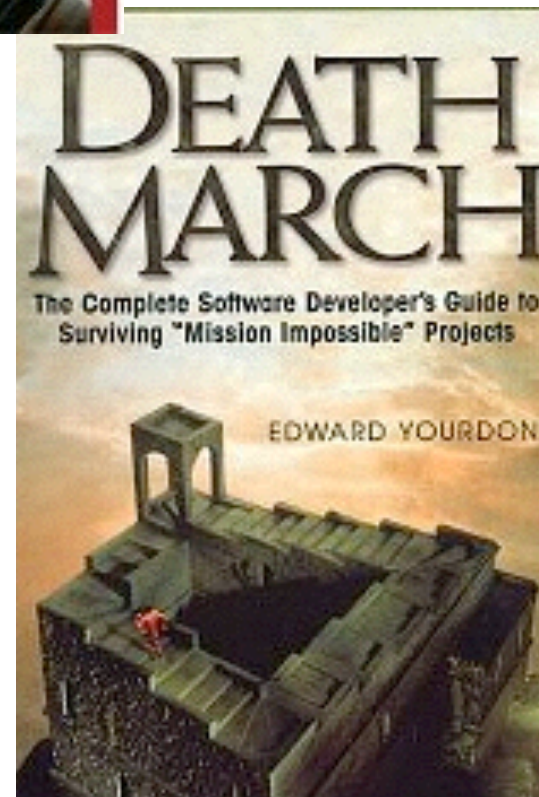
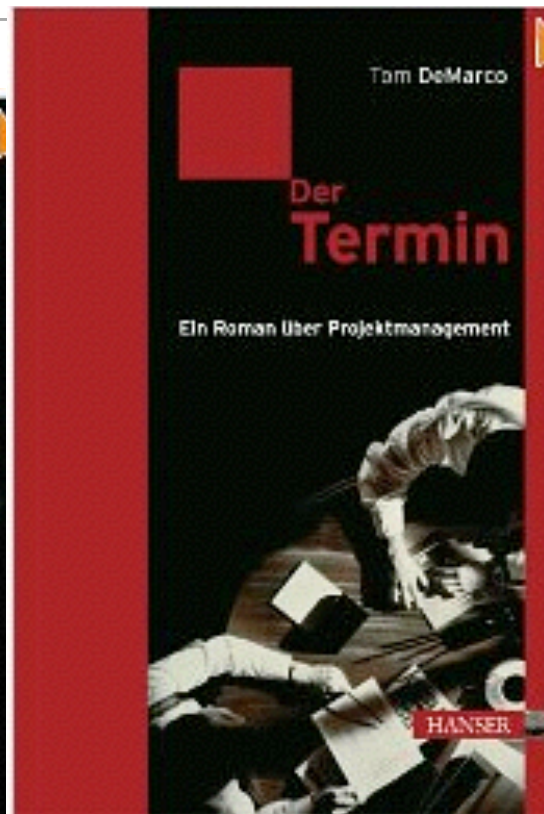
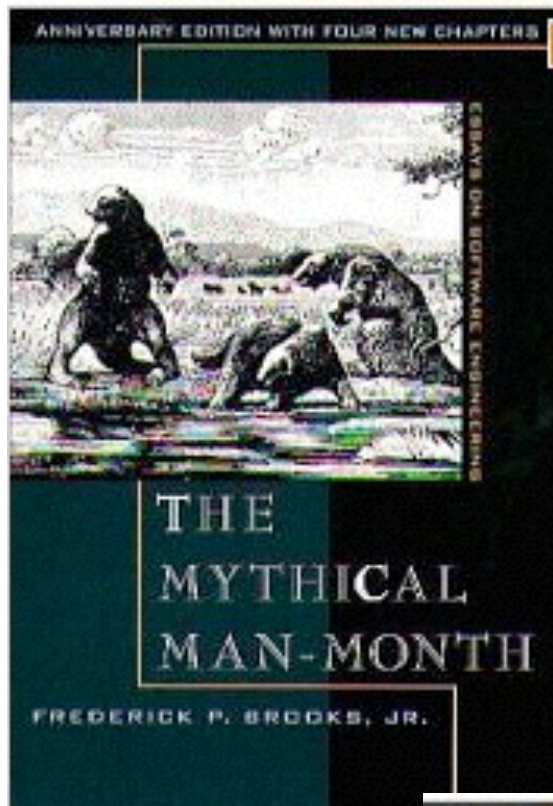


## Lesestunde





# Einige Bücher mit Bezug zu „weichen“ Erfolgsfaktoren

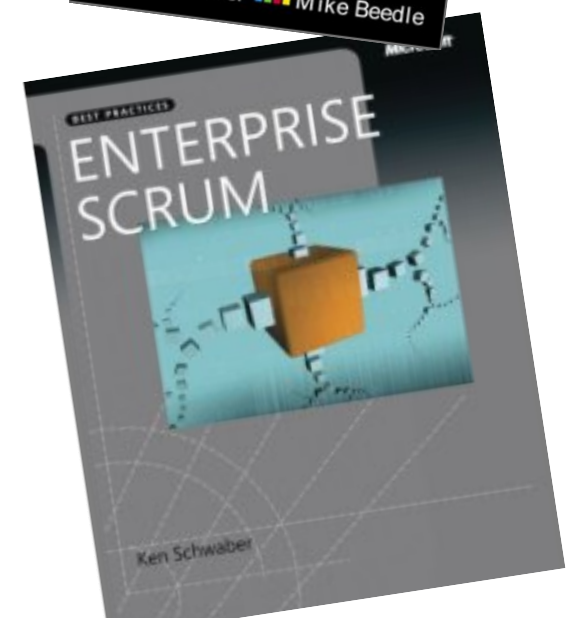




# Ein paar Bücher zu Scrum

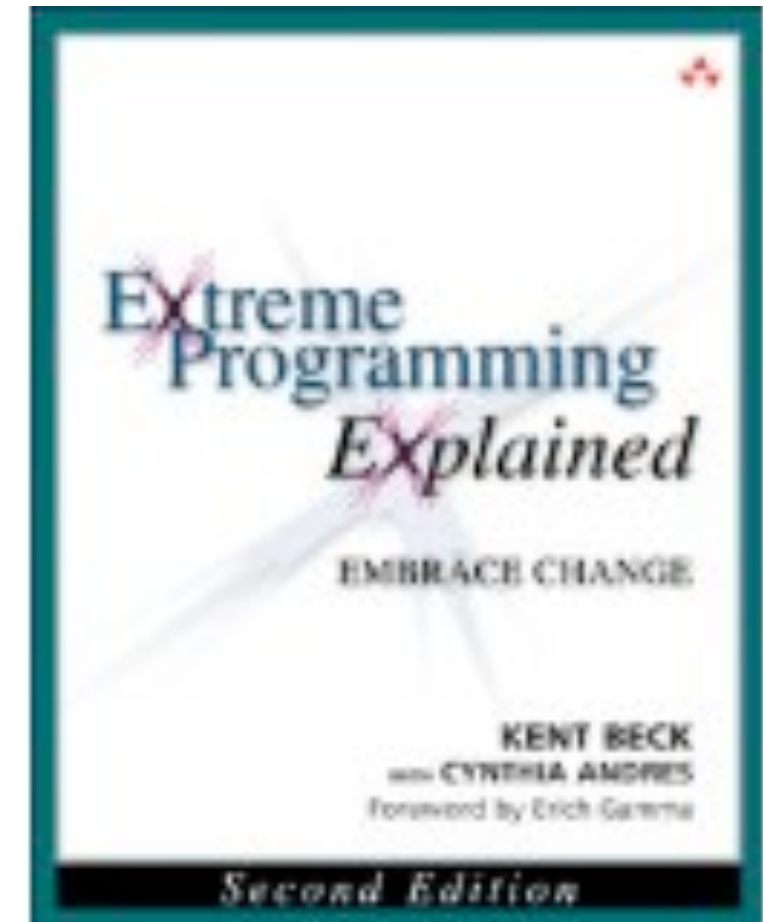
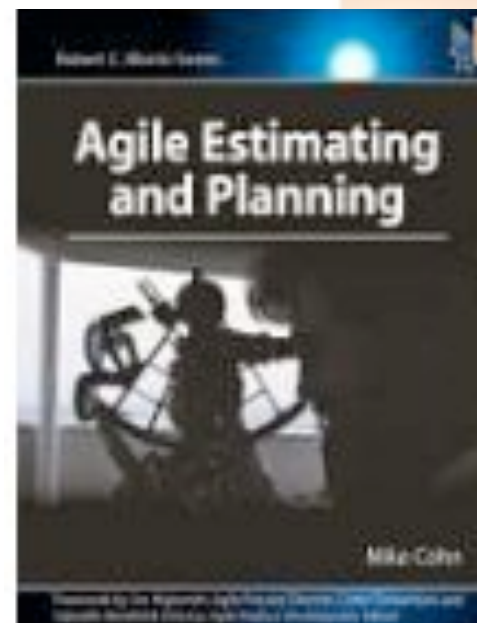
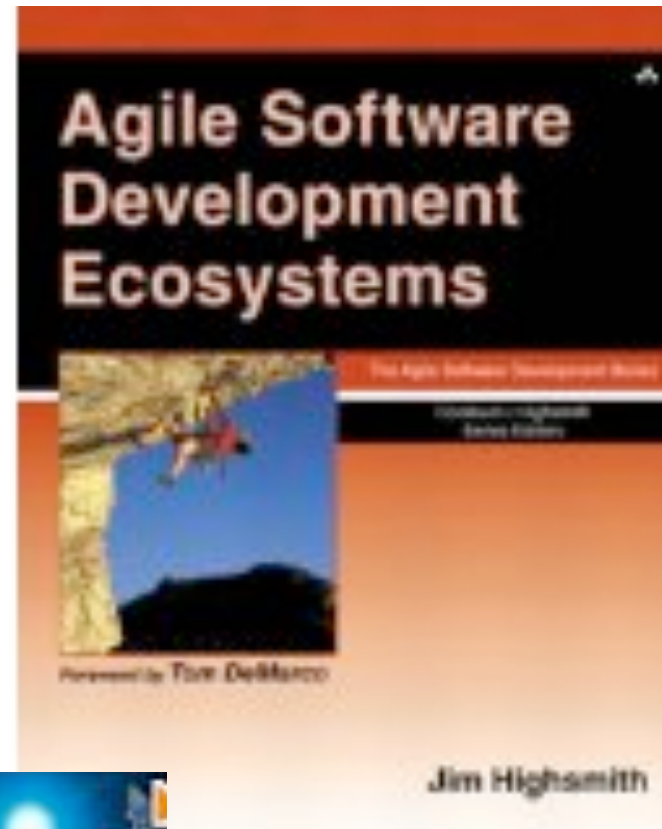
---

- Jeff Sutherland
  - Initiale Scrums bei Easel Corp., 1993
- Ken Schwaber
  - Präsentierte mit Sutherland Scrum auf OOPSLA 96
  - Autor von drei Büchern über Scrum
- Mike Beedle
  - Präsentierte Scrum-Pattern auf der PLOPD4
- Ken Schwaber und Mike Cohn
  - Scrum Alliance in 2002 gegründet



# Bücher über Agile Methoden

---





# Ausblick & Fragen

---

- Heute
  - Wrap Up Einflussfaktoren
  - Agile
- Planspiel
- Prüfungsvorbereitung



Thank You!

# Links & Literature

---

- Links
  - [www.scrumalliance.org](http://www.scrumalliance.org)
  - [www.it-agile.de/scrum-screencast.html](http://www.it-agile.de/scrum-screencast.html) (Video)
  - [video.google.com](https://video.google.com) „Scrum et al.“ (Video von Ken Schwaber)



# Bildnachweis

---

- Alle nicht explizit genannten Fotos/Grafiken von J. Pechau
- Alle Buchcover von Amazon.de, Logos von PMI und Scrum Alliance
- „Rugby Scrum“ by Paolo Camera, Flickr
- „?“ by florianmarquardt, Flickr
- SampleBurndownChart.png by Pablo Straub
- „Makray Memorial Golf Club, Barrington, Illinois“, Flickr, by danperry.com
- „Standup Meeting“, Flickr, by sjbrodwal
- „Busy Sprint“, Flickr, by drewgstephens

