Projektmanagement

...und was ist mit Agile?

Teil 13 - Projektmanagement - WS 2012/13

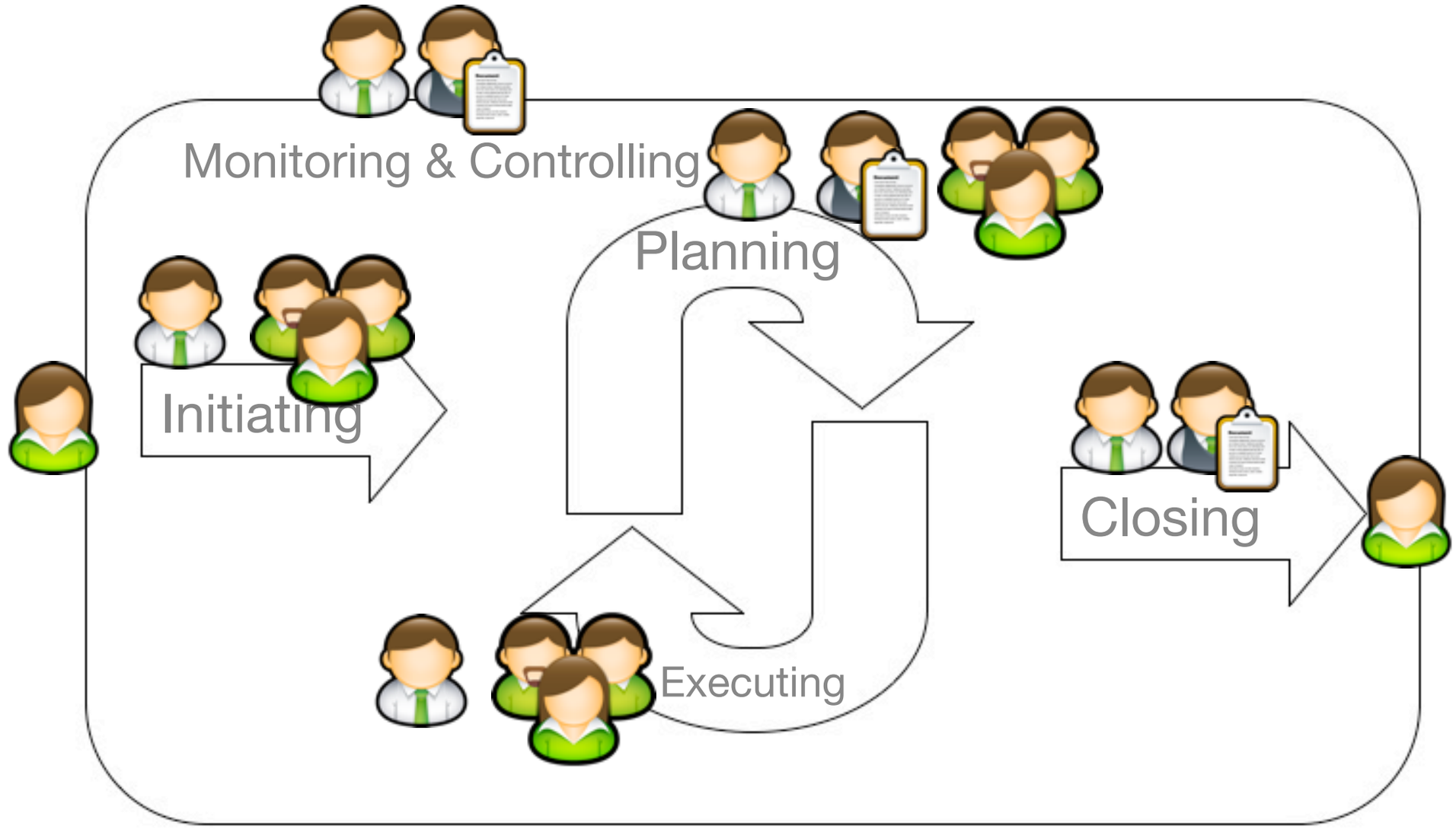
Jörg Pechau
Department Informatik, Uni Hamburg

Agenda

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

Kurze Erinnerung

Prozessgruppen des Projektmanagements



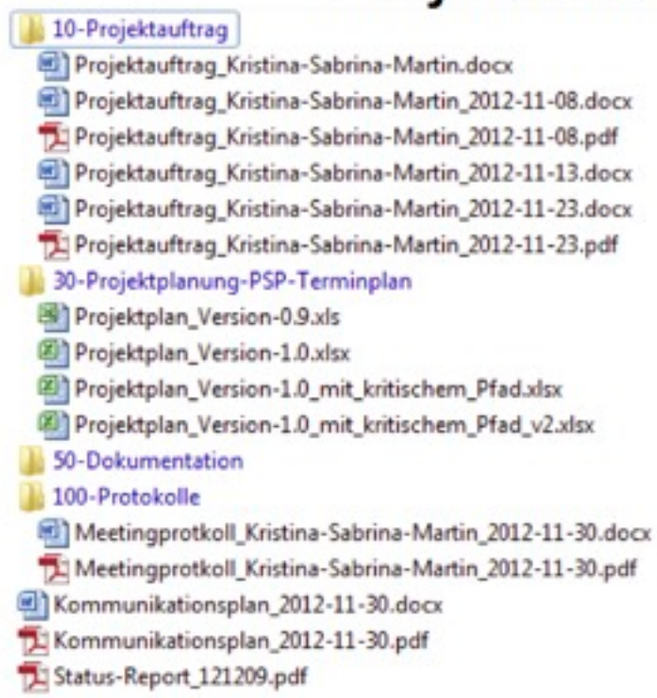
- **Die Projektmanagementprozesse lassen sich in 5 Gruppen zusammenfassen.**
- Die sequentielle Darstellung ist eine vereinfachende Schematisierung, die Prozesse wiederholen sich und interagieren mit einander.
- Diese Darstellung entspricht dem [PMBoK], die anderen Standards gruppieren die Prozesse entsprechend.

Lösungen

Projektakte

- Transparenz wo möglich,
  - z.B. Pläne, Zwischenstände etc.
- Vertraulichkeit wo nötig,
  - z.B. Wirtschaftlichkeitsberechnungen, Personenbezogene Daten, IPR etc.
- Kombination aus physischen und digitalen Artefakten

Projektakte - Struktur



- Führende Ziffern: Einteilung nach Priorität
- Gleiche Ordnerstruktur (phy. / digital): bessere Übersicht

Konflikte

- Meistens bekommen wir zwischenmenschliche Probleme über Symptome mit, das zu erkennen und dann noch an der Ursache zu arbeiten ist
  - Extrem schwer
  - Zeichen eines reifen Teams, wenn Konflikte offen angesprochen werden

(„Weiche“) Erfolgsfaktoren

„Wrap Up“

Erinnerung: Die „harten“ Erfolgsfaktoren

- Unterstützung durch Management
- Klarer Projektauftrag, erreichbares Ziel mit ebenso klaren Kriterien zur Zielerreichung
- Professionelles Projektmanagement
  - Belastbare, realistische Planung („Plan“)
  - Steuerung („Monitor & Control) vor allem
    - Reporting und Analysen
    - Kosten-, Risiko- und Änderungsmanagement
- Professionelle Umsetzung („Execute“)

Erinnerung: Die „harten“ Erfolgsfaktoren

- Unterstützung durch Management
- Klarer Projektauftrag, erreichbares Ziel mit ebenso klaren Kriterien zur Zielerreichung
- Professionelles Projektmanagement
- Belastbare, realistische Planung („Plan“)
- Steuerung („Monitor & Control) vor allem
  - Reporting und Analysen
  - Kosten-, Risiko- und Änderungsmanagement
- Professionelle Umsetzung („Execute“)

Projektarbeit heißt für und mit Menschen arbeiten, deswegen sind die schwer messbaren, so genannten „weichen“ Erfolgsfaktoren wichtig!

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Projektkultur**
  - Soll angstfrei & motivierend sein
  - Bearbeitung Konflikten, Feedback & Kritik wird offen ausgetauscht
  - Wird beeinflusst durch Erfahrungen des Teams bzw. seiner Mitglieder u.a. mit anderen Kulturen
  - Wird beeinflusst durch die Kultur der Projektumgebung
- **Führung** (siehe Vorlesung Nr. 7)
  - Führen und folgen, ist in einem „gesunden“ Team ein Wechselspiel
  - Basiert auf klaren Rollen und Verantwortungen - fix wo nötig und selbstorganisiert wo möglich
  - Führungsstil sollte partizipativ sein, wenn überhaupt nur in Krisen auf „Kommandostil“ zurückgreifen

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Projektkultur**
  - Soll angstfrei & motivierend sein
  - Bearbeitung Konflikten, Feedback & Kritik wird offen ausgetauscht
  - Wird beeinflusst durch Erfahrungen des Teams bzw. seiner Mitglieder u.a. mit anderen Kulturen
- **Führung** (siehe Vorlesung Nr. 7)
  - Führen und folgen, ist in einem „gesunden“ Team ein Wechselspiel
  - Basiert auf klaren Rollen und Verantwortungen - fix wo nötig und selbstorganisiert wo möglich
  - Führungsstil sollte partizipativ sein, wenn überhaupt nur in Krisen auf „Kommandostil“ zurückgreifen

Nicht aufgearbeitete Konflikte ruinieren auf Dauer jede Kultur, erschweren die Führung, zerstören Vertrauen und gehen auf Kosten der Motivation!

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Kommunikation** (siehe frühere Vorlesung)
  - Muss offen und transparent sein
  - Zeitnah erfolgen und auf den Punkt sein
  - Soll informieren und nicht in erster Linie unterhalten

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Kommunikation** (siehe frühere Vorlesung)
  - Muss offen und transparent sein
  - Zeitnah erfolgen und auf den Punkt sein
  - Soll informieren und nicht in erster Linie unterhalten

Die einzelnen Erfolgsfaktoren nicht komplett voneinander unabhängig!

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Vertrauen**
  - Bekommt man nicht geschenkt, muss man sich verdienen
  - Ist schwer erreicht und leicht verspielt
- Ist Basis für eine effektive Zusammenarbeit, beispielsweise weil z.B.
  - Risiken und Fehler benannt werden, d.h. gelernt und agiert werden kann
  - Weniger „gemanaged“ und „überwacht“ werden muss und
  - Weniger Zeit in Politik und Absicherung geht...

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Vertrauen**
  - Bekommt man nicht geschenkt, muss man sich verdienen
  - Ist schwer erreicht und leicht verspielt
- Die einzelnen Erfolgsfaktoren nicht komplett voneinander unabhängig!
  - Risiken und Fehler benannt werden, d.h. gelernt und agiert werden kann
  - Weniger „gemanaged“ und „überwacht“ werden muss und
  - Weniger Zeit in Politik und Absicherung geht...

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Motivation**
  - Durch ein Ziel das allen bekannt ist und auf das sich alle „comitten“ können
  - Jeder kennt seinen Aufgabenbereich und traut sich diesen zu
  - Alle benötigen Aufmerksamkeit und Anerkennung
  - Gerechtigkeit und Fairness motiviert nicht unbedingt,
  - „Abwesenheit“ von Gerechtigkeit und/oder Fairness demotiviert zuverlässig
- **Einmal mehr: Mitdenken hilft... :-)**

Zusammenfassung: Die „weichen“ Erfolgsfaktoren

- **Motivation**
  - Durch ein Ziel das allen bekannt ist und auf das sich alle „comitten“ können
  - Jeder kennt seinen Aufgabenbereich und traut sich diesen zu

Dies ist eine „**best of**“ Liste, es gibt wesentlich mehr „weiche“ Erfolgsfaktoren!

  - Gerechtigkeit und Fairness motiviert nicht unbedingt,
  - „Abwesenheit“ von Gerechtigkeit und/oder Fairness demotiviert zuverlässig

- **Einmal mehr: Mitdenken hilft... :-)**

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.), Continous 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 Software Development (2001)

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

http://agilemanifesto.org

Andere Agile Methoden (Auszug der Wichtigsten)

- Software Entwicklungsmethoden
  - XP - eXtreme Programming
  - BDD - Behaviour Driven Development
  - TDD - Test Driven Development
  - Lean Software Development
  - Crystal
- (Projekt-) Managmentmethoden
  - Kanban
  - SCRUM

Andere Agile Methoden (Auszug der Wichtigsten)

- Software Entwicklungsmethoden
  - XP - eXtreme Programming
  - BDD - Behaviour Driven Development
  - TDD - Test Driven Development
  - Lean Software Development
  - Crystal

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

- (Projekt-) Managmentmethoden
  - 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.

http://manifesto.softwarecraftsmanship.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 Änderungswahrscheinlickeit, z.B. explorative Projekte
  - Projekte mit klarem statischen Scope gewinnen durch „Agile“ häufig nichts oder nicht viel
- Häufiges Feedback hilft
  - Optimalen Kundennutzen zu erreichen
  - Risiko durch Fehlentwicklung zu reduzieren
  - Probleme in Organisaton und Aufgabe frühzeitig aufzuzeigen

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 starken statischen Scope gewinnen durch „Agile“ häufig nichts oder nicht viel

- Häufiges Feedback hilft

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

- Optimalen Kundennutzen zu erreichen

- Risiko durch Fehlentwicklung zu reduzieren

- Probleme in Organisaton 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
    - Eigenverantwortung
    - Autonome Teams
- Fokus auf Erreichung des optimalen Nutzwerts für Kunden

Scrum ist empirisches Management

- Kernideen
  - „Inspect and adapt“
  - Selbstorganisation
  - Selbstmanagement
  - Eigenverantwortung
  - Autonome Teams
- Fokus auf Erreichung des optimalen Nutzwerts für Kunden

Erfordert Transparenz, den Mut loszulassen und Änderungswillen!

Rollen

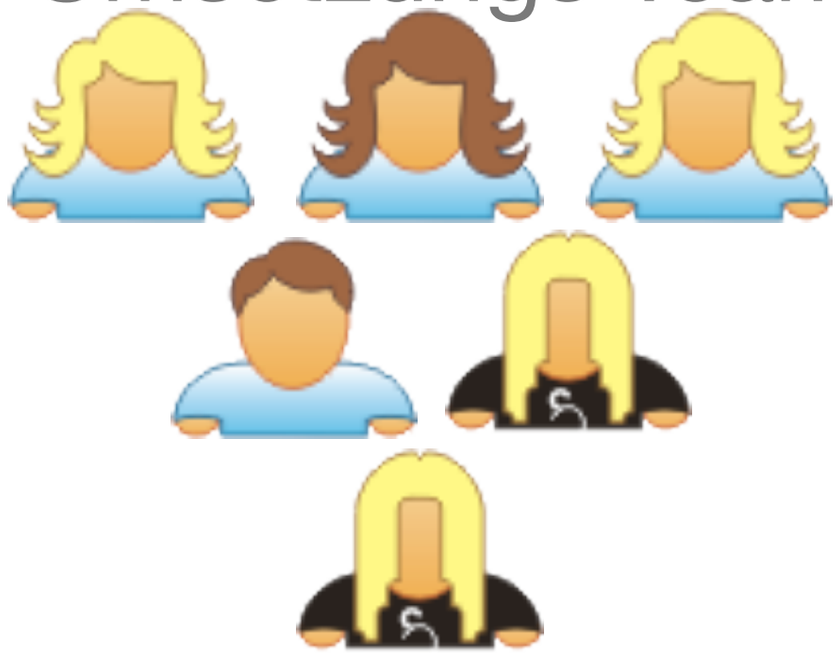
Projektorganisation

Das Scrum Team

Product Owner

Scrum Master

Umsetzungs Team

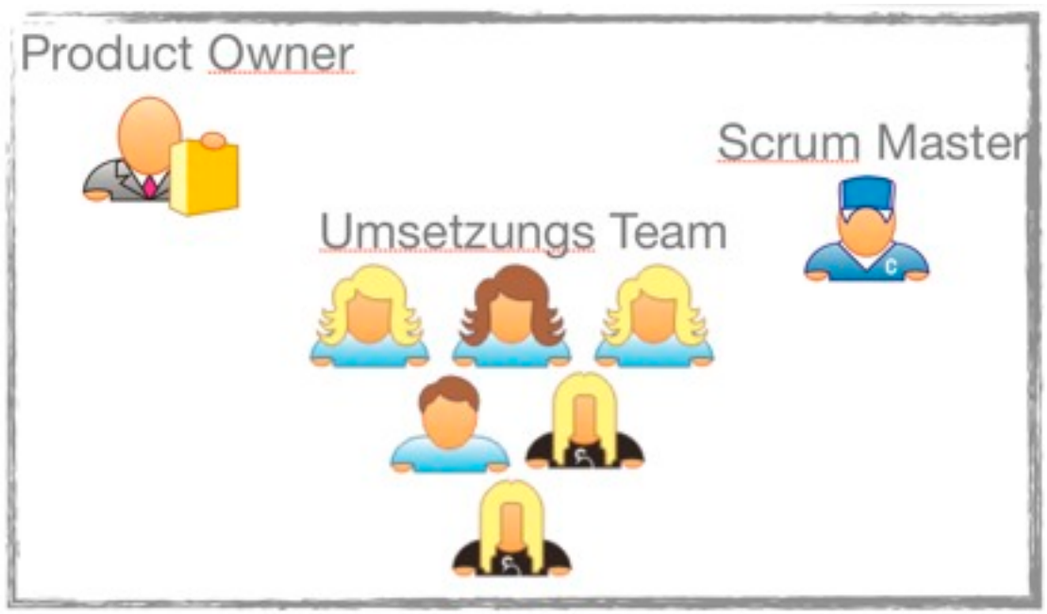


Projektorganisation

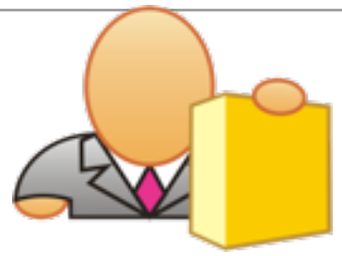
Kunden

Management

Scrum Team



Der Product Owner



- Repräsentiert den Kunden, bzw. ist der Kunde

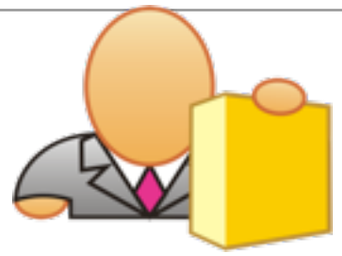
- Definiert das Produkt
  - Beschreibt den Funktionsumfang in Form von Epen, Themen, User Storys
  - Sammelt diese im Product Backlog
  - Pflegt und priorisiert das Product Backlog

- Bestimmt Release-Planung und Inhalt des Produkts
  - Priorisiert Features, z.B. abhängig vom Marktwert
  - 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 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
  - Priorisiert Features, z.B. abhängig vom Wert
  - Passt Features und Prioritäten nach Bedarf für jeden Sprint an

Der Product Owner trägt die inhaltliche und kommerzielle Verantwortung!
Er ist kein Projektleiter!

- 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
  - Versucht Raum für Kreativität und Selbstorganisation zu steigern
  - Verbessert die engineering practices
  - 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
  - Kümmert sich um Hindernisse (Impediments / Impediment List)

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
  - Verbessert die engineering practices

Der Scrum Master verantwortet den Prozess ist aber kein Projektleiter!

- 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
- Kümmert sich um Hindernisse (Impediments / Impediment List)

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
  - „Commited“ sich

- 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

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

Product Backlog

- Priorisierte Liste von „Product Backlog Items“
  - Sammlung der funktionaler und nicht funktionaler Anforderungen
  - Unterschiedliche Granularität
  - Form:
    - Epics, Topics, User Stories
- In der Sprint Planung „wandern“ Product Backlog Items in das Sprint Backlog
- Realisierungsaufwand wird in relativem, abstrakten Maß geschätzt

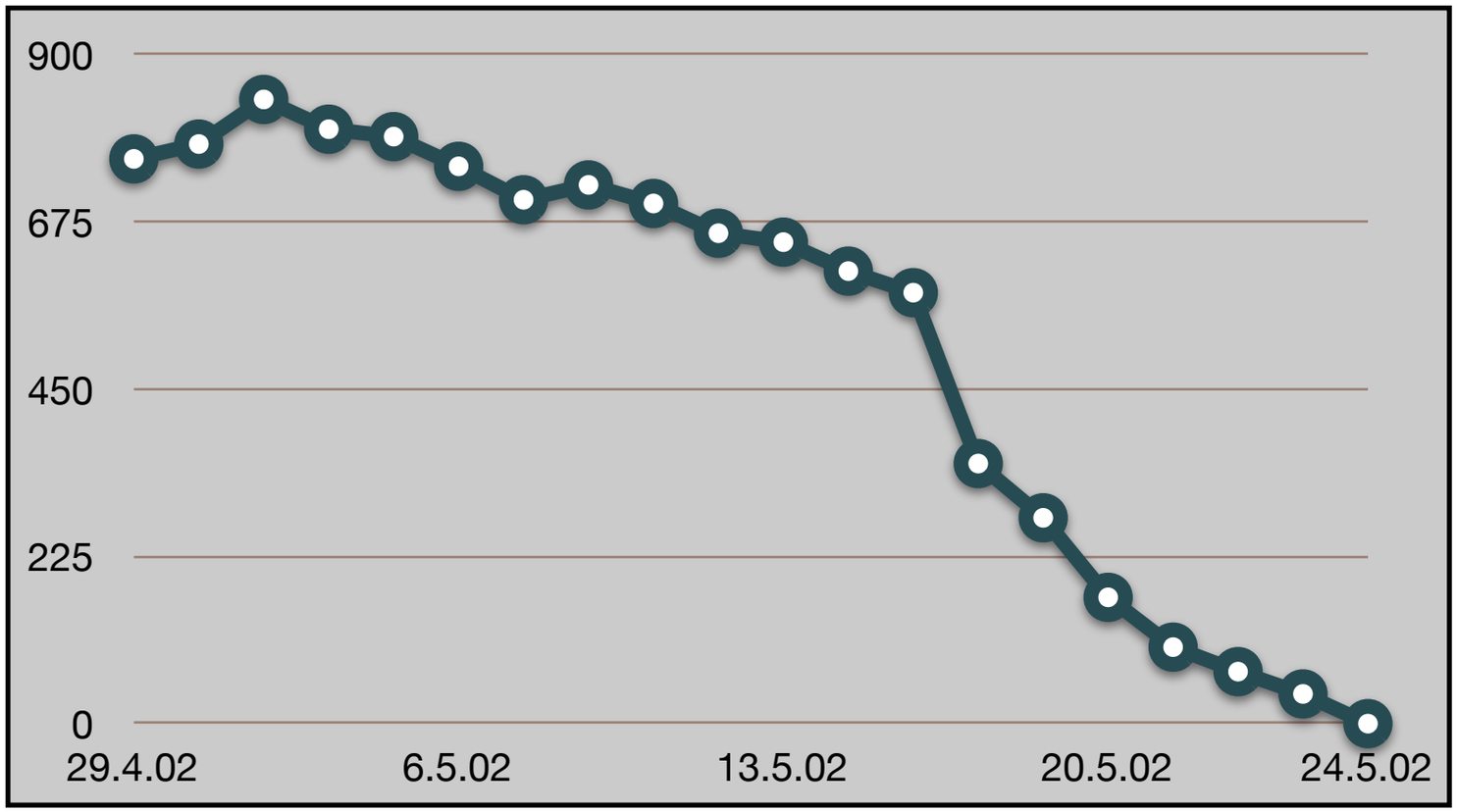
Viele können (evtl.) das Product Backlog ergänzen,
nur der Product Owner priorisiert es!

Sprint Backlog

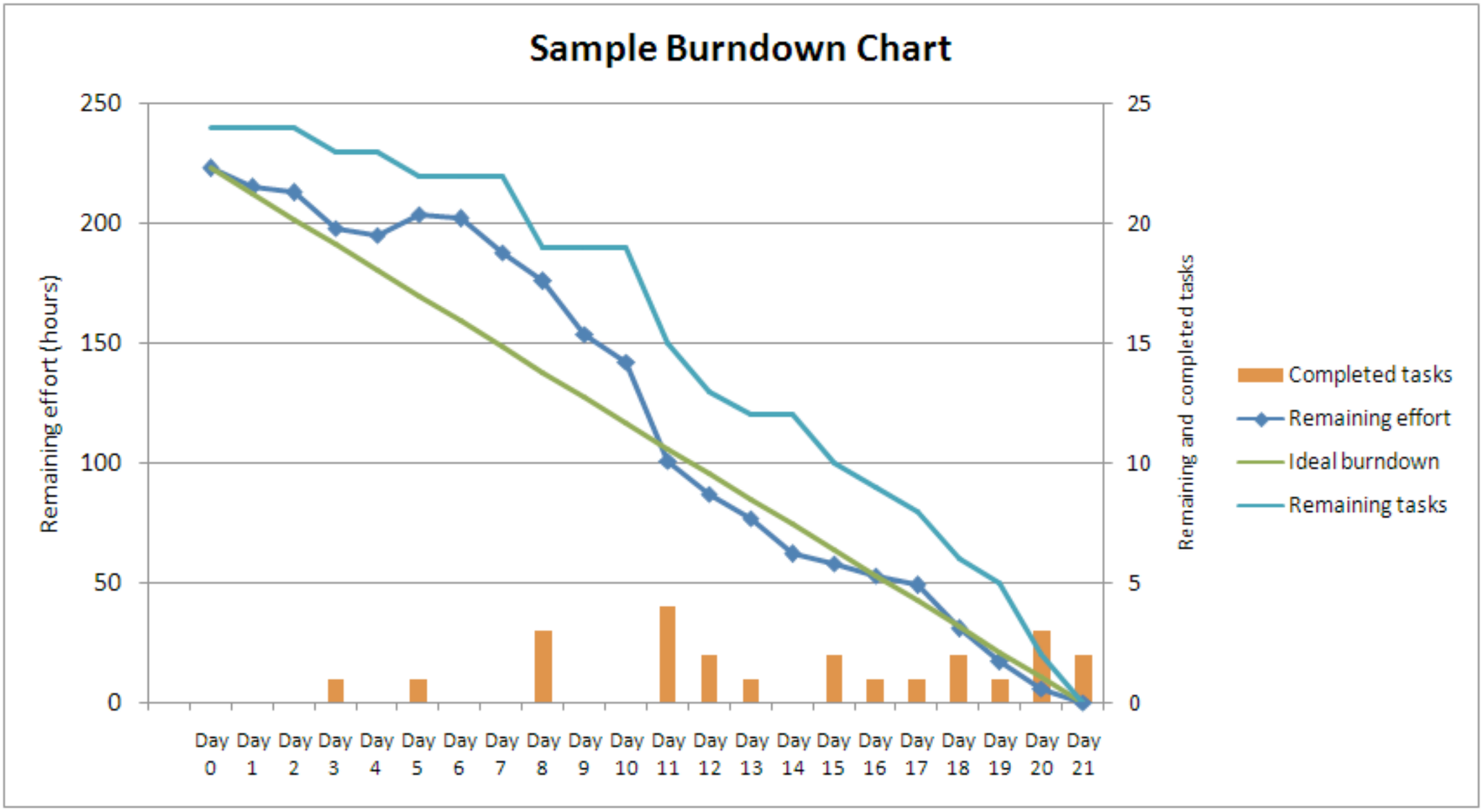
- Definiert die Aufgaben für einen Sprint
  - Um das Sprint Ziel zu erreichen und
  - Auf die sich das Team commited hat
- User Storys werden dazu in notwendige Tasks heruntergebrochen
- Tasks werden in Stunden geschätzt
- Während eines Sprints wird der Scope von außen nicht mehr geändert

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

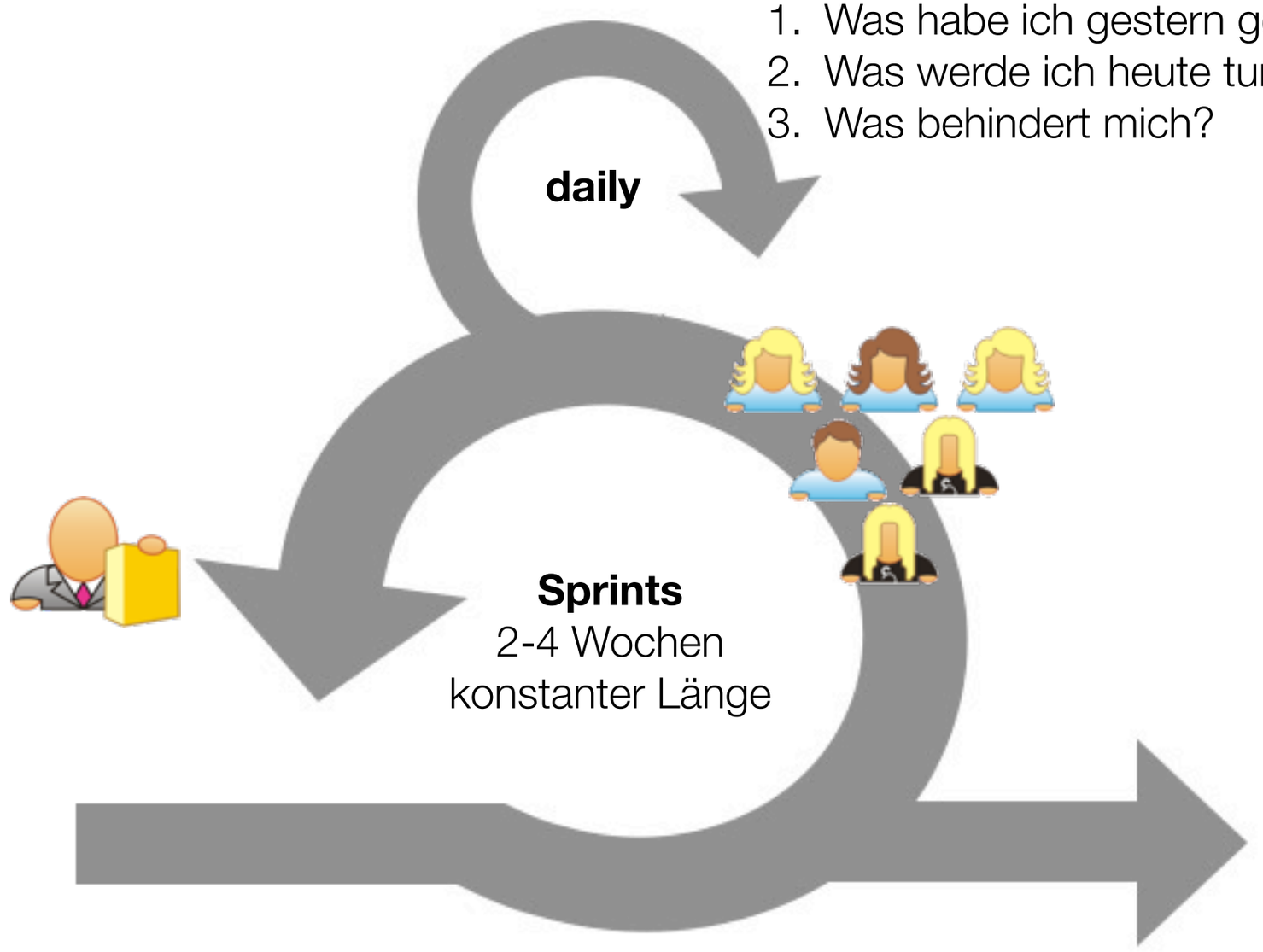
Wie läuft ein Scrum Projekt ab?

Generelles Vorgehen

- Ziel:
  - **Das „minimal viable Product“**
- 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

Daily Scrum
1. Was habe ich gestern getan?
2. Was werde ich heute tun?
3. Was behindert mich?



Product Backlog
Enthält die aktuell bekannten Anforderungen

Sprint Planung
Abbildung der Prioritäten auf das Machbare

Sprint Backlog
Anfoderungen für den nächsten Sprint

Sprint Ergebnis
Potenziell lieferbares Produkt Inkrement

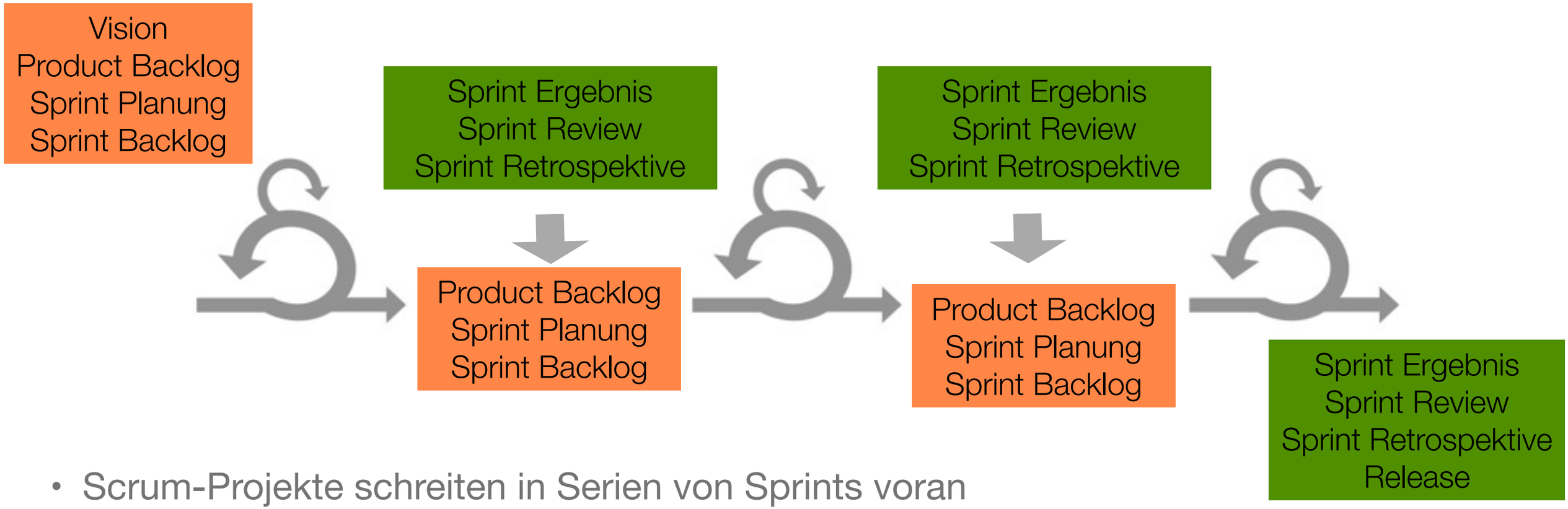
Sprint Review
Das Team präsentiert die Ergebnisse, Abnahme durch den Product Owner „Inspect and adapt“

Sprint Retrospektive
„Inspect and adapt“

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

- Product Backlog
- Sprint Review Ergebnisse
- Gelerntes aus Sprint Retrospektive
- Team Geschwindigkeit

- Sprint Priorisierung (Product Owner)
  - Product Backlog analysieren
  - Product Backlog ggf. neu priorisieren
- Sprint Planung (Team)
  - Aus Product Backlog (Epen, Themen, User Stories) das Sprint Backlog (Tasks)ableiten, um Sprint Ziel zu erreichen
  - Sprint Backlog schätzen
  - Team commited sich auf die Sprint Planung entsprechend Team-Geschwindigkeit (Velocity)
- Scrum Master unterstützt den Prozess

- Sprint Ziel / Definition of Done
- Sprint Backlog

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 Ware“,, Reports etc.!
- Product Owner beurteilt das Ergebnis anhand
  - Des Sprint-Ziels und
  - Der Definition of Done
- Scrum Master organisiert und stellt den Prozess sicher

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

Inspect & Adapt

- Stets das Produkt zeigen keine „Slide-Ware“, Reports 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 sich verbessern möchte, z.B.
  - „Beginnen mit...“
  - „Aufhören mit...“
  - „Weitermachen mit...“

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

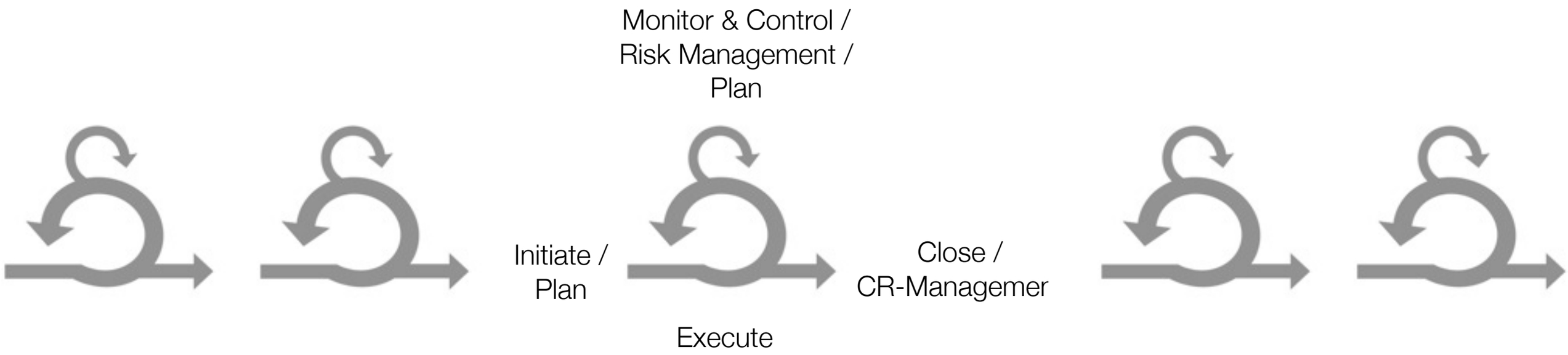
Inspect & Adapt

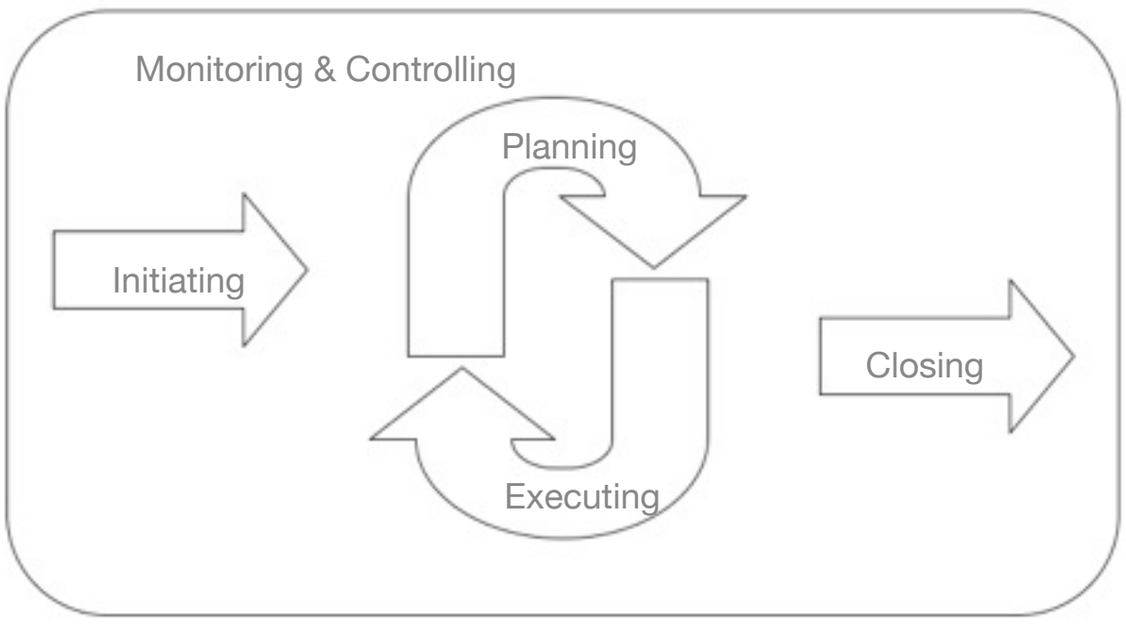
- Das Scrum Team diskutiert, wie es sich verbessern möchte, z.B.
  - „Beginnen mit...“
  - „Aufhören mit...“
  - „Weitermachen mit...“

Agiles und klassisch phasenrientiertes PM

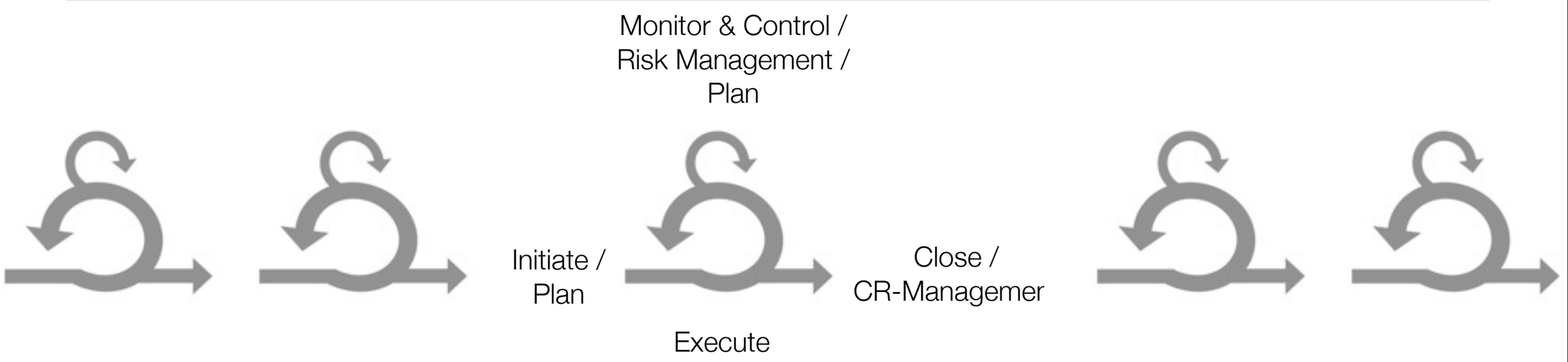
Eine grobe „Abbildung“

Ungefähres Mapping Prozessgruppen PM auf Scrum Prozess





Ungefähres Mapping Prozessgruppen PM auf Scrum Prozess



Vorgehen stellt Kunden und Änderungen in den Mittelpunkt!

Was macht Scrum so anders?

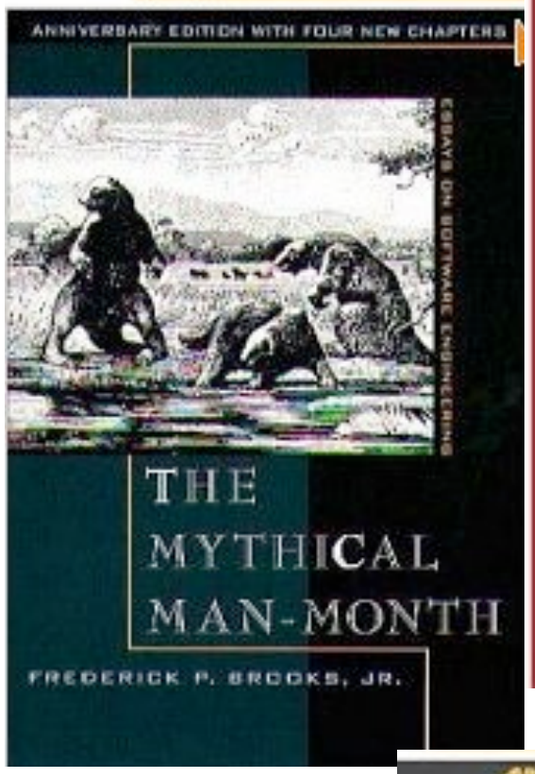
- Scrums Vokabular, Rollen, Artefakte und Prozesse sind in „10 Min“ erlernt
  - Mächtig, einfach, aber nicht simpel
  - Wirkt einfach, erfordert aber viel Disziplin
- Agiles Vorgehen
  - Gut bei explorativen Projekten, bzw. hoher Change-Wahrscheinlichkeit
  - Erfordert i.a. Haltungsänderung - das liegt nicht jedem...
- Agiles und nicht agiles Vorgehen lassen sich prinzipiell kombinieren, erfordert aber eine Reihe von Kompromissen

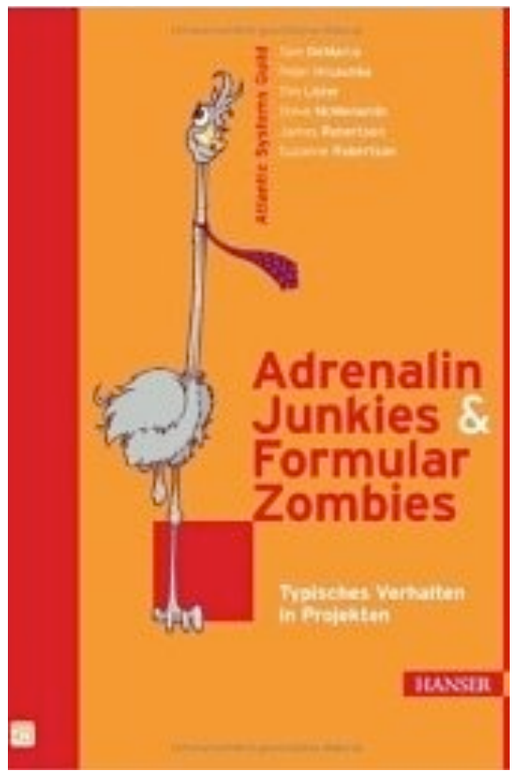
Was macht Scrum so anders?

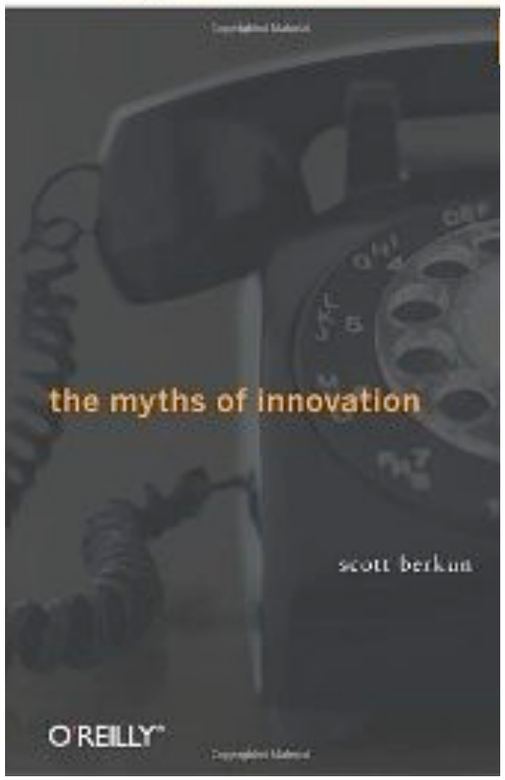
- Scrums Vokabular, Rollen, Artefakte und Prozesse sind in „10 Min“ erlernt
  - Mächtig, einfach, aber nicht simpel
  - Wirkt einfach, erfordert aber viel Disziplin
- Der eigentliche Unterschied zu klassisch phasenorientiertem Projektmanagement liegt im zugrundeliegenden Wertesystem!
  - Gut bei explorativen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

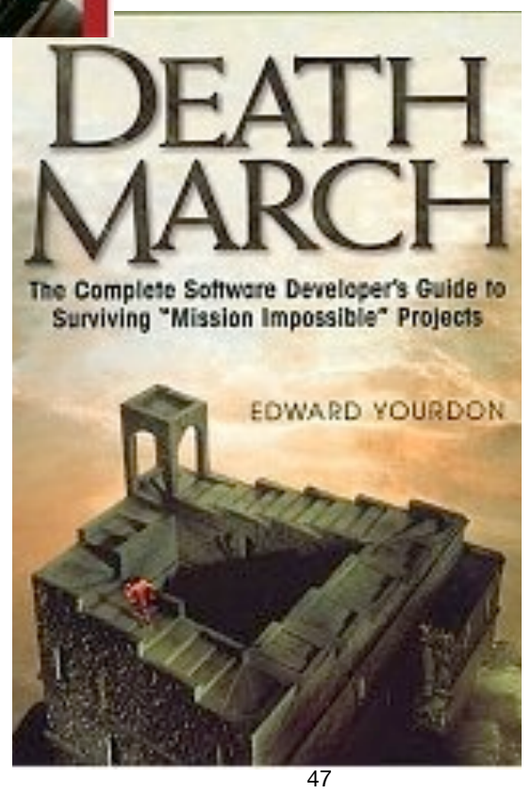
Lesestunde

Einige Bücher mit Bezug zu „weichen“ Erfolgsfaktoren





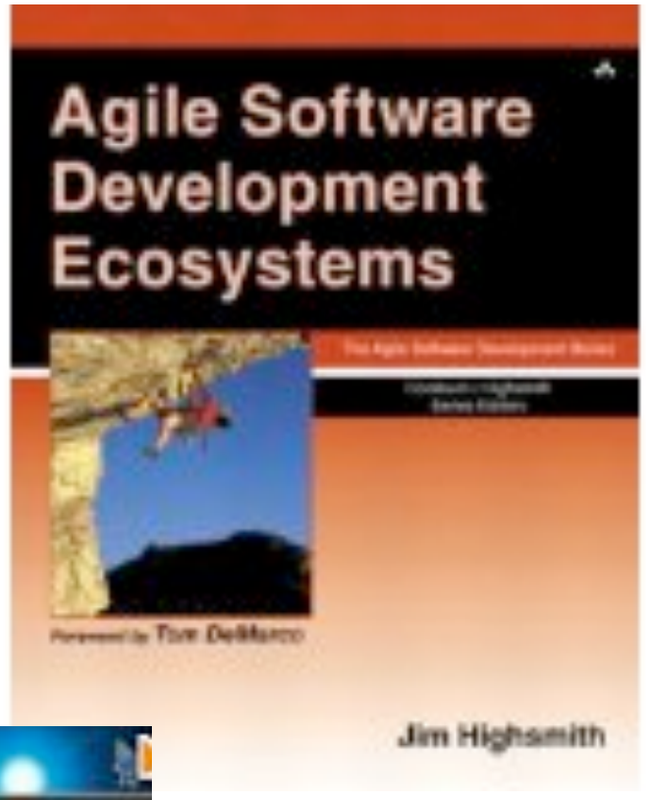


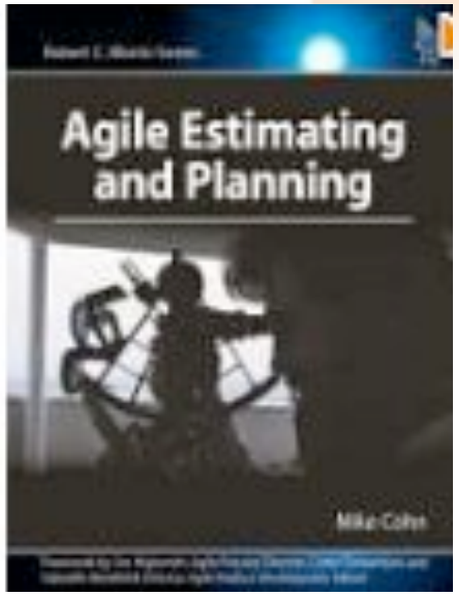


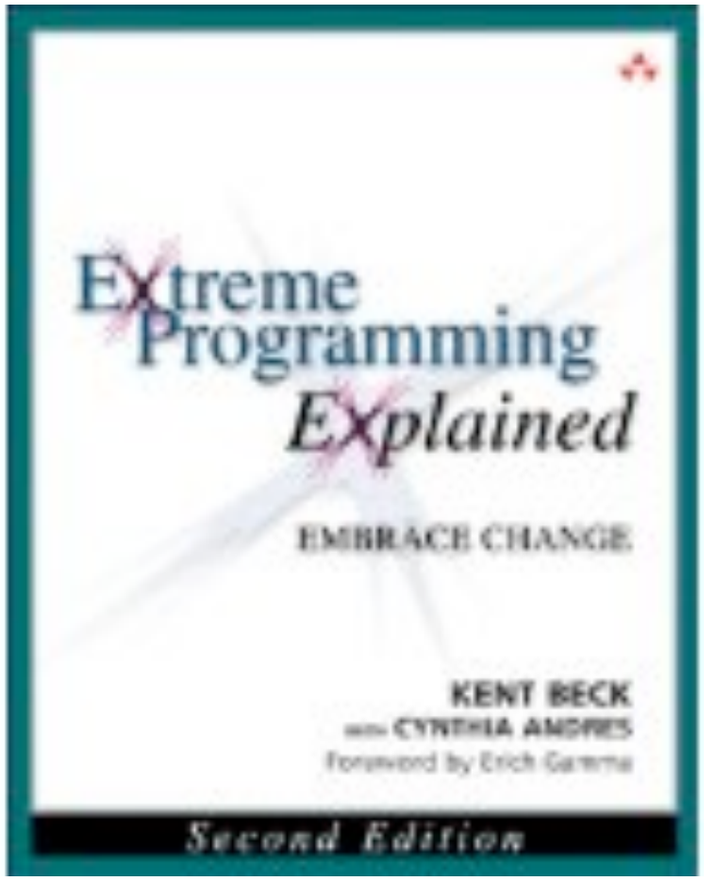
Ein paar Bücher zu Scrum

- 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

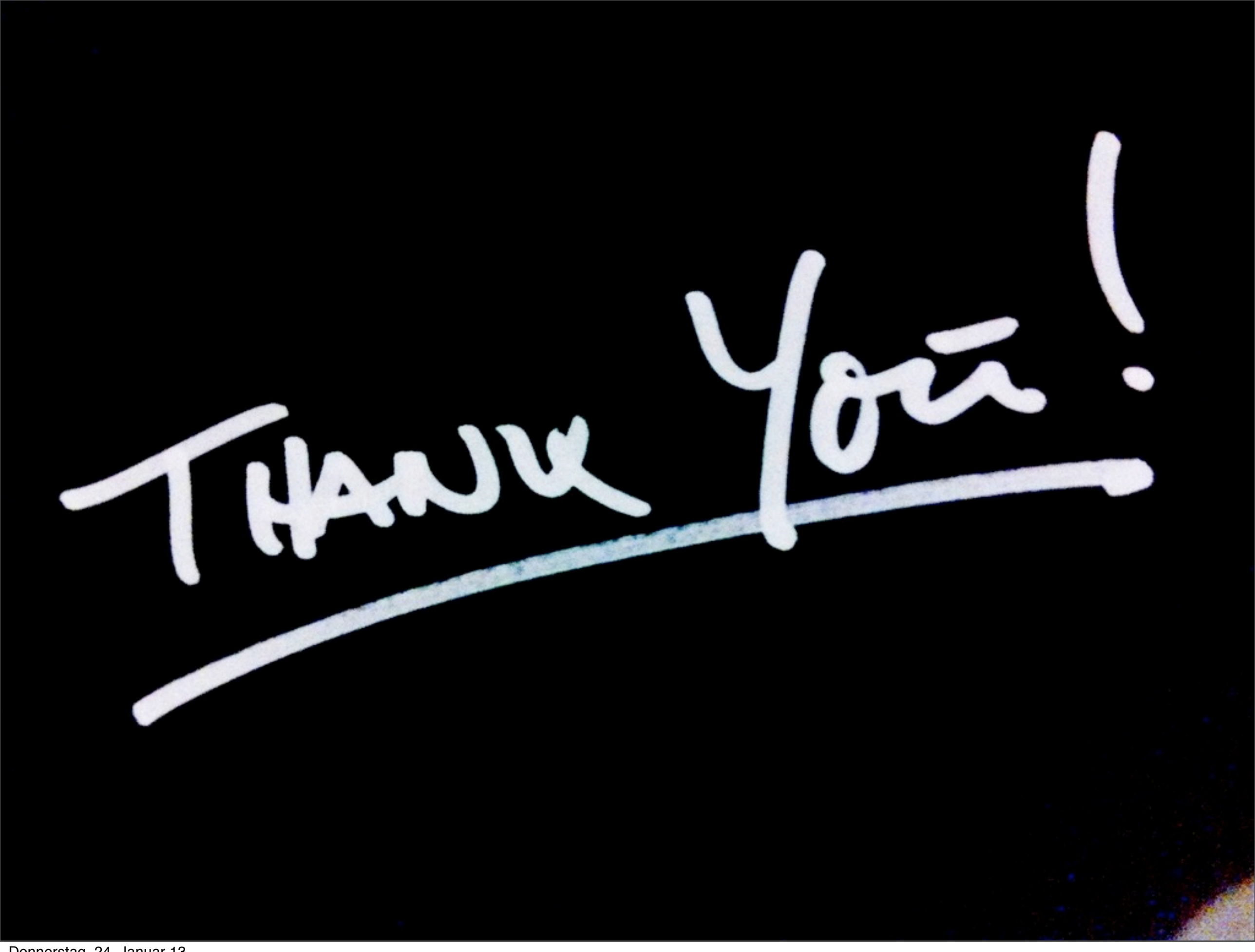






Ausblick & Fragen

- Heute
  - Wrap Up Einflussfaktoren
  - Agile
- Prüfungsvorbereitung


Thank You!

Links & Literature

- Links
  - www.scrumalliance.org
  - www.it-agile.de/scrum-screencast.html (Video)
  - 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