Technische Hochschule Rosenheim
Technical University of Applied Sciences

Projektmanagement

Sommersemester 2020

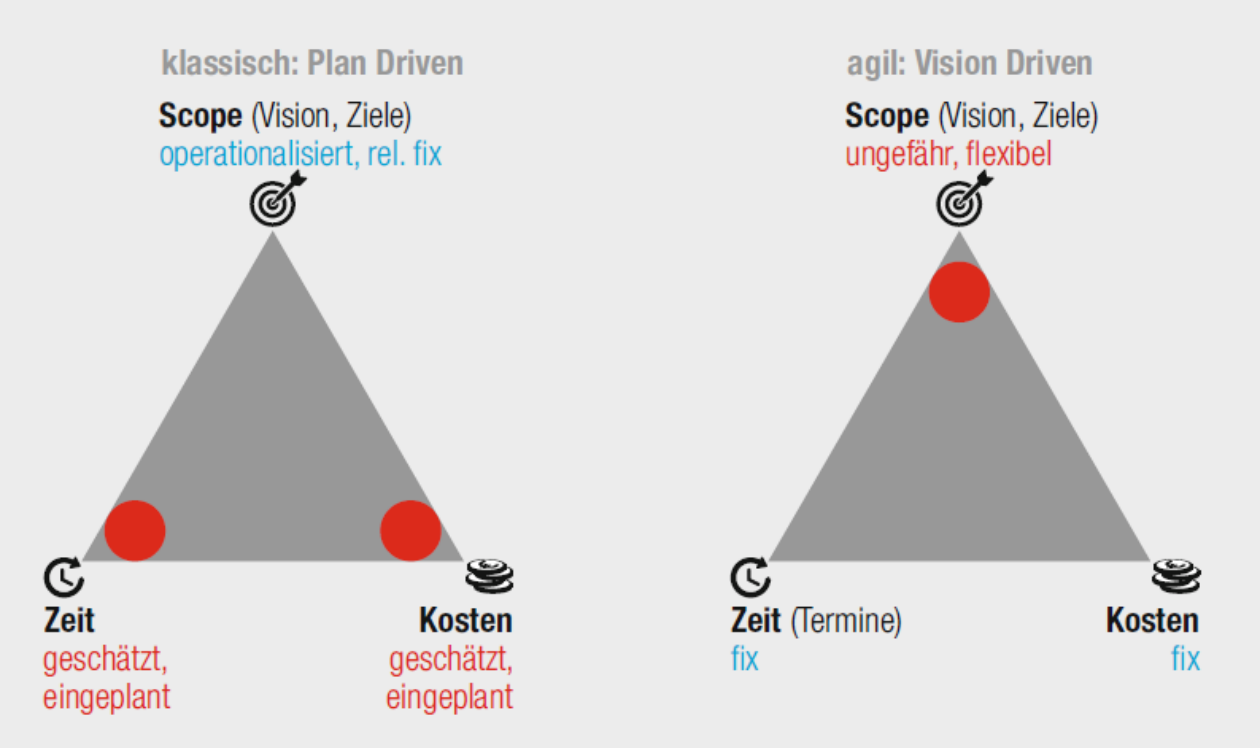
Prof. Dr. Claudia Förster / Prof. Dr. Ewald Jarz

Vorlesungsüberblick

1. Einführung
2. Klassisches Projektmanagement
3. Agiles Projektmanagement
4. Hybrides Projektmanagement

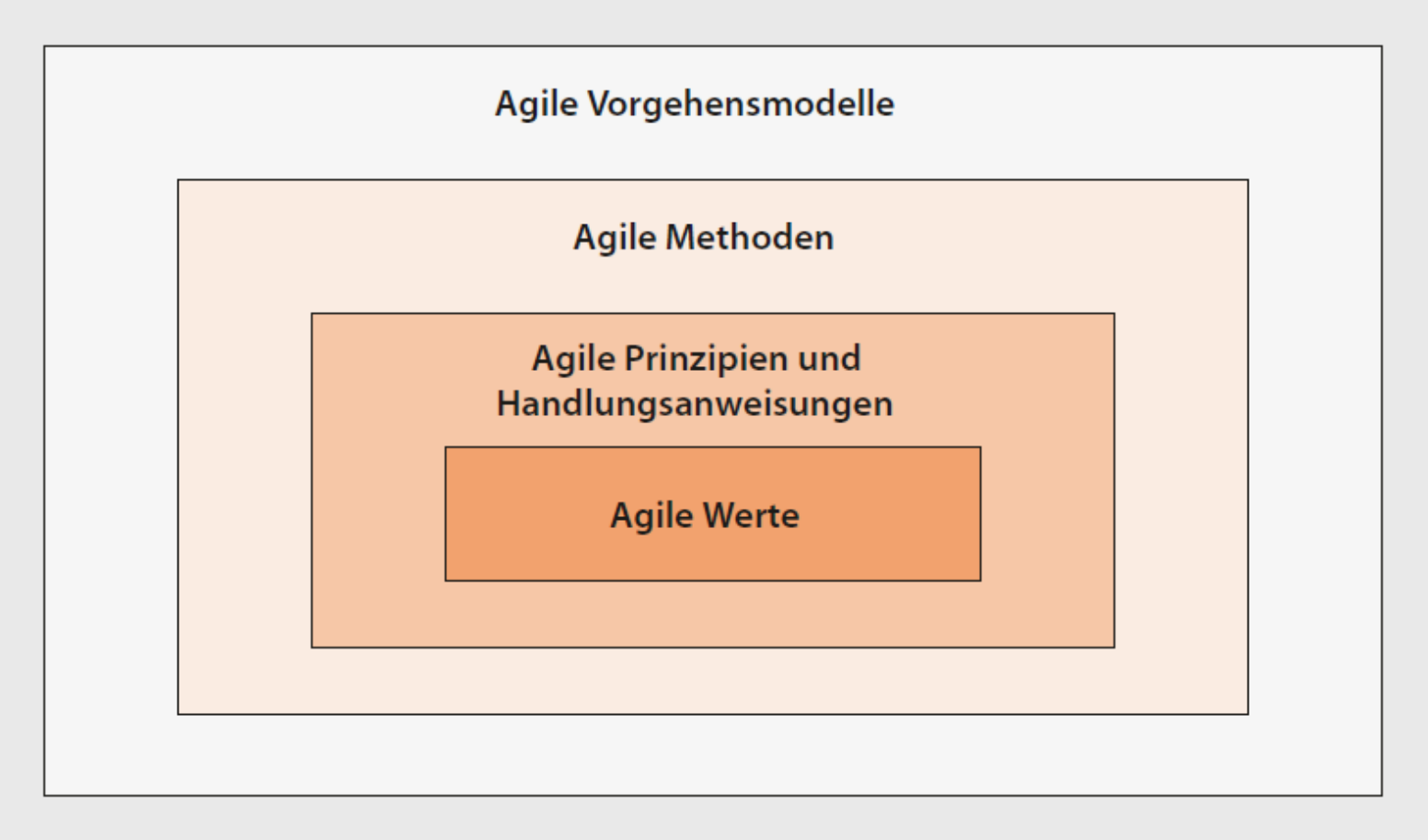
Agiles Projektmanagement (1/2)

- Bewegliches, flinkes, reflexives und lernendes Vorgehen
- Magische Dreieck im klassischen und agilen Projektmanagement



Quelle: Kuster, et.al. (2019): Handbuch Projektmanagement, S. 81

Agiles Projektmanagement (2/2)



Quelle: Dechange, A. (2020): Projektmanagement schnell erfasst, S. 286

Wertaussagen aus dem agilen Manifest

Individuen und Interaktionen	wichtiger als	Prozesse und Tools
Laufende Software	wichtiger als	Ausführliche Dokumentation
Zusammenarbeit mit dem Kunden	wichtiger als	Vertrags-verhandlungen
Reagieren auf Veränderungen	wichtiger als	Planbefolgung

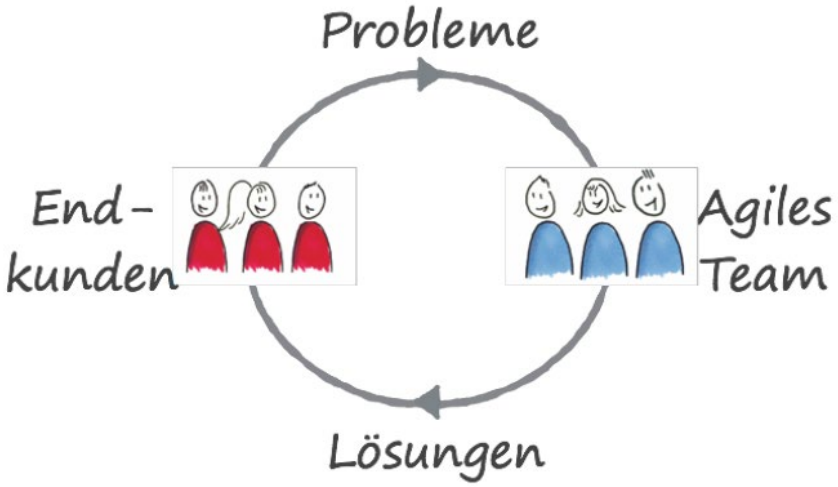
Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 8

Prinzipien des agilen Manifests

Prinzipien agilen Projektmanagements	Häufige Praxis in traditionellen Projekten
■ Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen. ■ Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.	■ Traditionelle Projekte sind plangetrieben: Der Kunde teilt zu Projektbeginn mit, was er haben möchte. Der spezifizierte Projektgegenstand wird dann den erstellten Plänen entsprechend umgesetzt.
■ Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.	■ Das plangetriebene Vorgehen führt zu einer negativen Einstellung gegenüber Änderungen. Bei Änderungen am Projektgegenstand müssen auch bereits existierende Pläne geändert werden.
■ Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten. ■ Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.	■ Entwicklerteams werden während der Projektlaufzeit neu zusammengestellt oder arbeiten an mehreren Projekten gleichzeitig. Durch sich ändernde Zuständigkeiten kann das Verantwortungsgefühl für den Projektgegenstand leiden. Gegenmaßnahme: stärkeres Controlling.
■ Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.	■ Traditionelle Vorgehensmodelle fokussieren auf Abläufe. Im Arbeitsalltag ersetzen E-Mails teilweise das persönliche Gespräch.
■ Funktionierende Software ist das wichtigste Fortschrittsmaß.	■ Der Projektfortschritt wird häufig in Kosten oder Mitarbeiterauslastung gemessen.
■ Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.	■ In der unternehmerischen Praxis werden viele Projektmanager mit großer Verantwortung und wenig Befugnissen ausgestattet. Dies fördert Projektverzug und Eskalationen.
■ Einfachheit - die Kunst, die Menge nicht getaner Arbeit zu maximieren - ist essenziell. ■ Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.	■ Komplexität soll durch Pläne beherrschbar gemacht werden, was nicht immer gelingt. Späte Änderungen oder Abstriche in der Qualität können die Folge sein.
■ Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams. ■ In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.	■ Klassisches Instrument des Wissensmanagements in traditionellen Projekten sind die Lessons Learned und die Projektanalyse am Projektende. Regelmäßige Reflexionen sind eher die Ausnahme.

Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 163

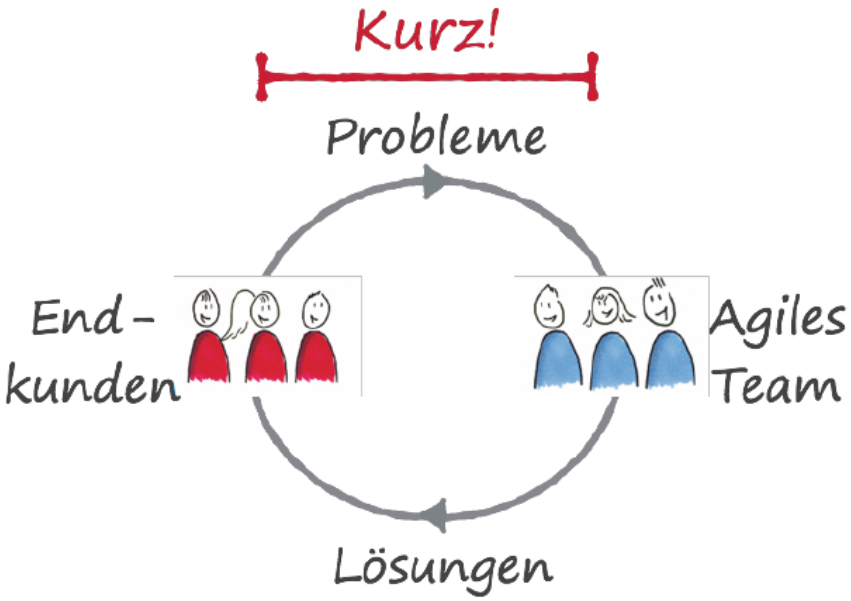
Agile Kernideen (1)



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 4

- Kundenrelevante Probleme lösen
- Direkter Kundenkontakt
  - Probleme und Bedürfnisse verstehen
  - Team findet angemessene Lösungen
  - Team liefert direkt Lösung

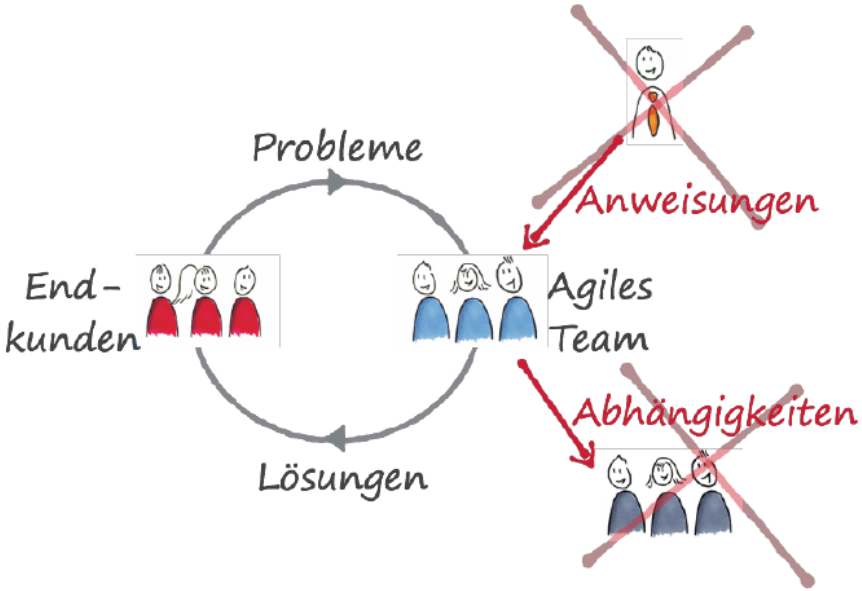
Agile Kernideen (2)



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 4

- Schnelle Reaktion
  - Problem-Lösungszyklus sollte schnell und häufig durchlaufen werden
  - Kunde bekommt schnell Lösungen
  - Zwischenstände zeigen und Feedback einholen

Agile Kernideen (3)



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 4

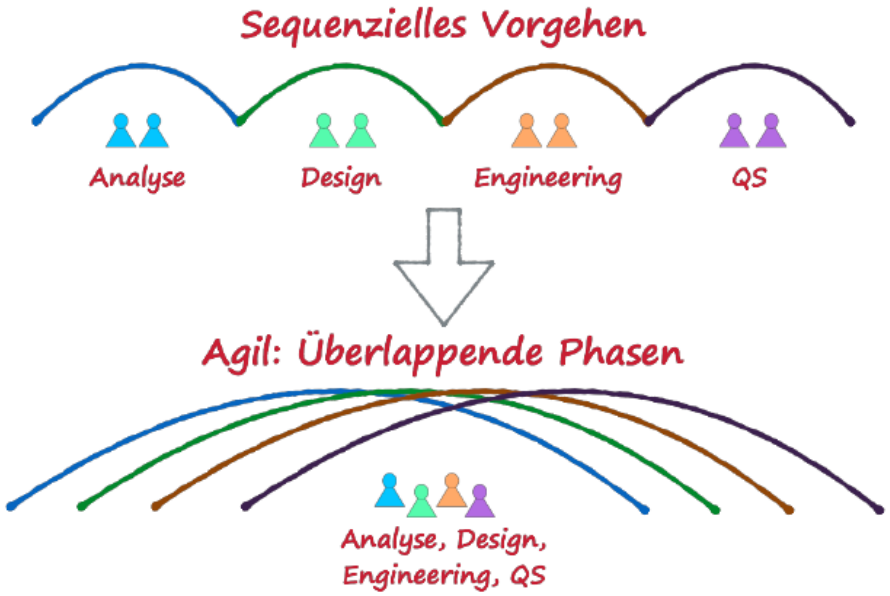
- Autonome und selbstorganisierte Teams
  - Keine Anweisungen von außen, wie es sich zu organisieren hat
  - Nicht von anderen Teams abhängig

Agile Kernideen (4)

Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 4

- Interdisziplinäre Besetzung
  - Große Bandbreite an Fähigkeiten notwendig
  - Spezialisierungen abhängig von Kunden und ihren Problemen

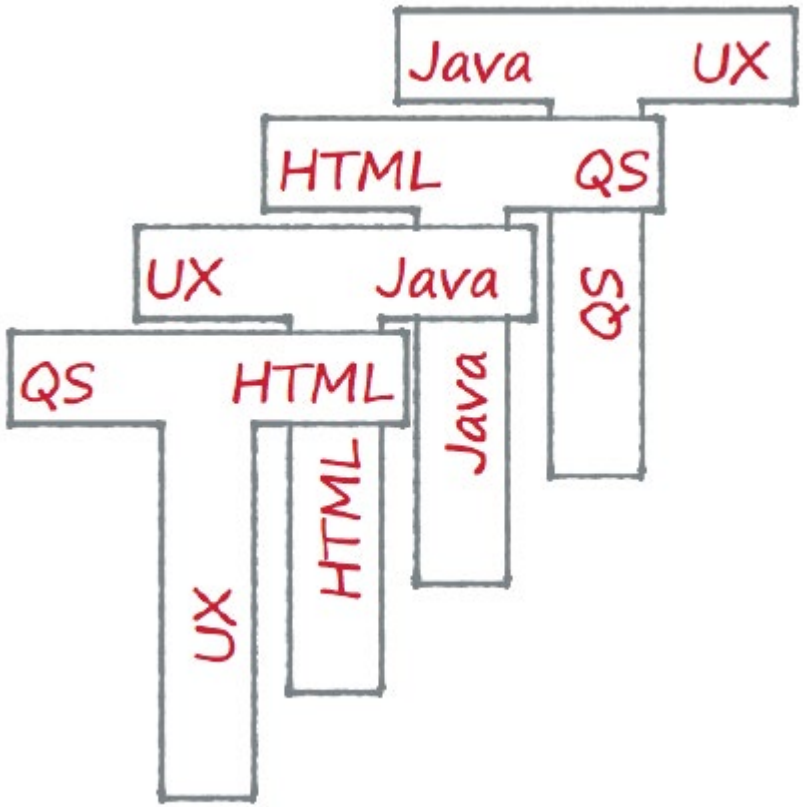
Agile Kernideen (5)



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 5

- Überlappende Phasen
  - Nicht nacheinander, sondern gleichzeitig Durchführung
  - Überlappung der Phasen hängt vom Kontext bzw. Dynamik ab

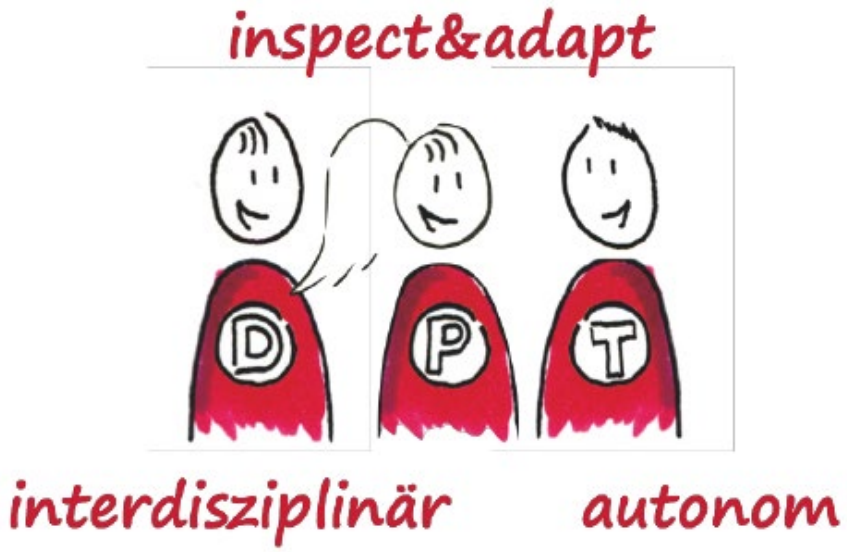
Agile Kernideen (6)



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 5

- T-Shaped-Skill-Sets
  - Spezialsierungen sollen und müssen erhalten bleiben
  - Zusätzliche Fähigkeiten
  - Überlappender Skill-Set trägt dazu bei, dass Arbeit im Fluss bleibt

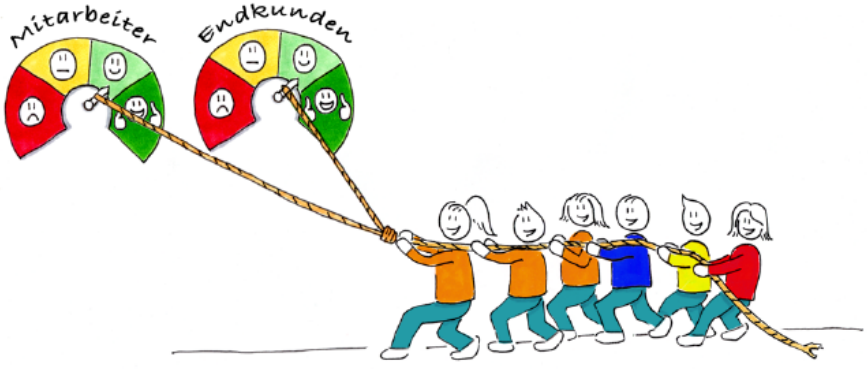
Zusammenfassung agiles Team



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 5

- Es ist autonom und arbeitet selbstorganisiert.
- Es ist interdisziplinär besetzt.
- Es verbessert Ergebnis und Prozess über Inspect&Adapt.

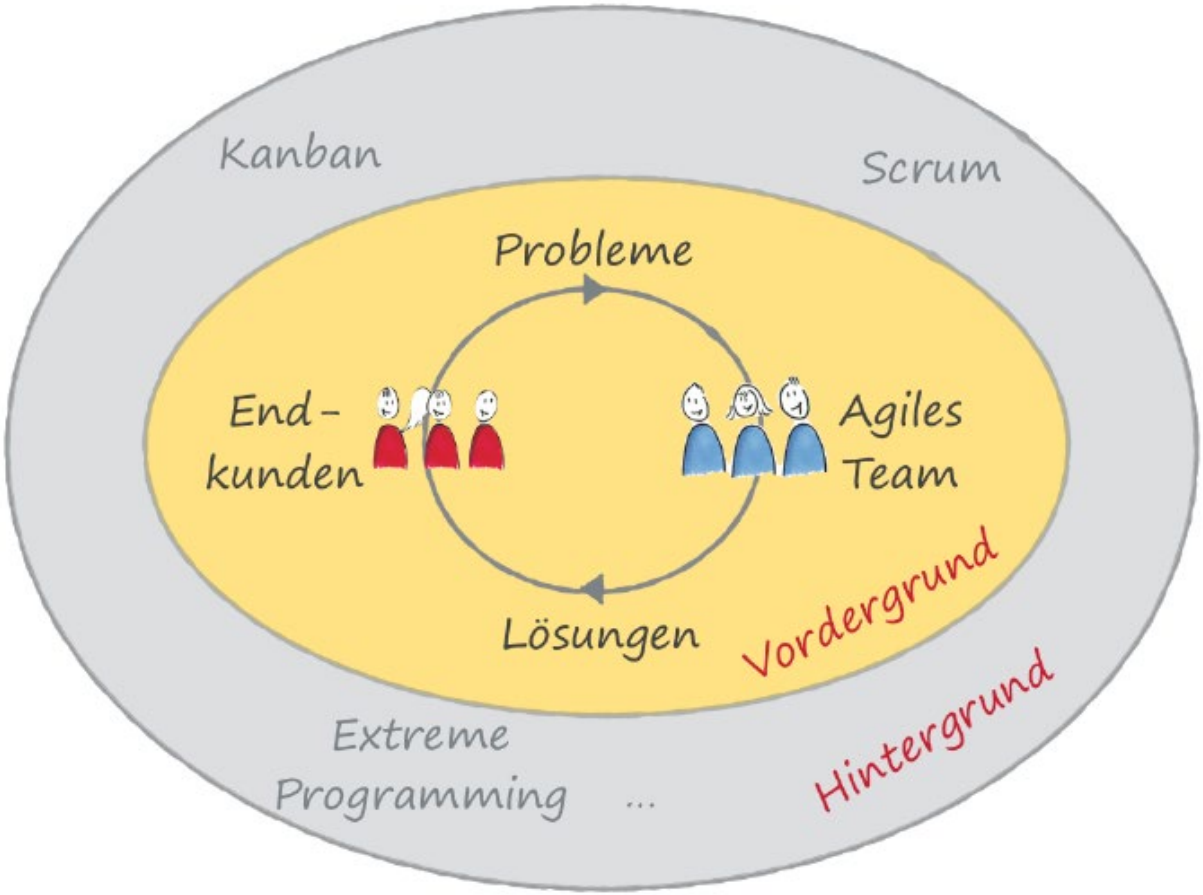
Vorteile von Agilität



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 7

- Überlappende Phasen => verkürzte Time-to-Market
- Inkrementell-iterative Vorgehen => Schnelle Reaktion
- Kundennähe => Angemessenere Lösungen für den Kunden
- Cross-funktionale Teams =>Höhere Wahrscheinlichkeit für Innovation
- Direkter Kundenkontakt und –feedback: mehr Sinn in der Arbeit

Methodische Ausgestaltungen agilen Arbeitens



Quelle: Roock, S. (2018): Agilität – eine Einführung, S. 10

Scrum – Definition und Erfinder

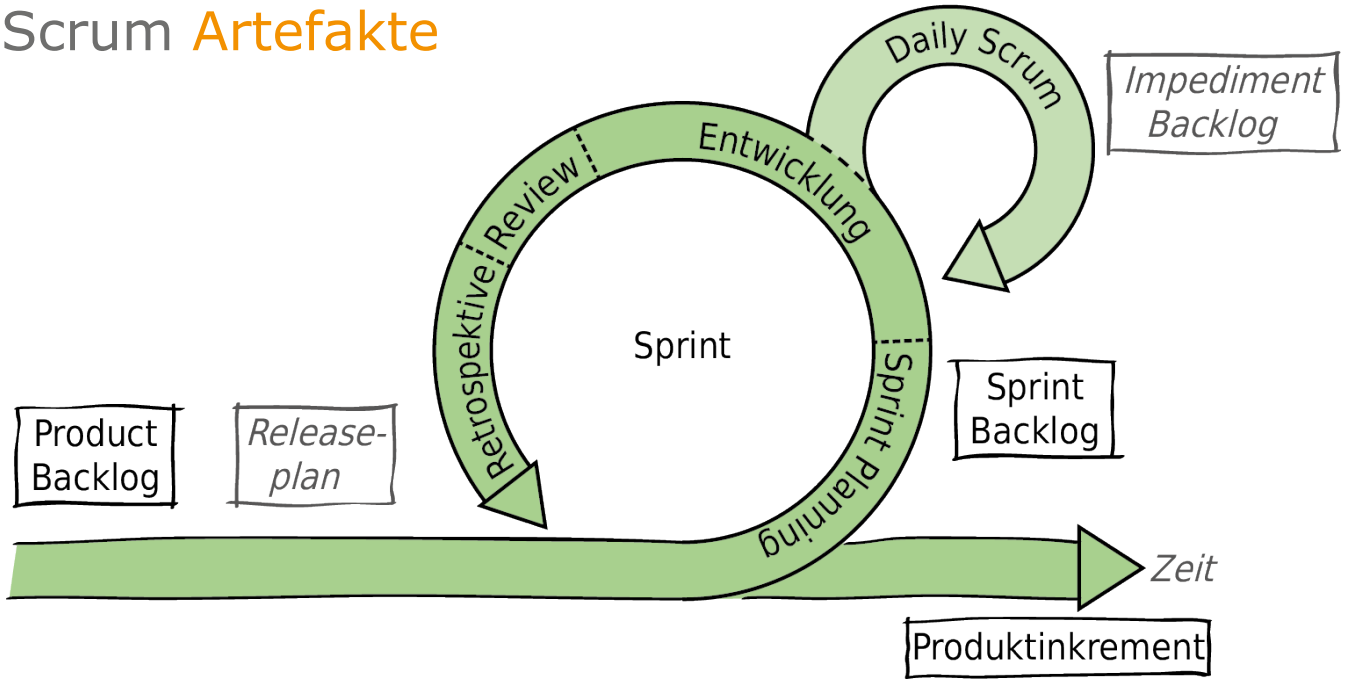
- Produktmanagement-Framework
  - Entwicklung, Lieferung und Weiterentwicklung von komplexen Produkten
  - Produktive und kreative Zusammenarbeit
  - Produkte mit maximalen Wert
- Ursprung des Ausdrucks:
  - Beim Rugby verwendete Strategie, bei der eine adaptive, schnelle, selbst organisierende Vorgehensweise mit wenigen Ruhepausen angestrebt wird

- Scrum Guide (*http://www.scrumguides.org/*)

Jeff Sutherland Ken Schwaber

Scrum-Prozess und wichtige Komponenten

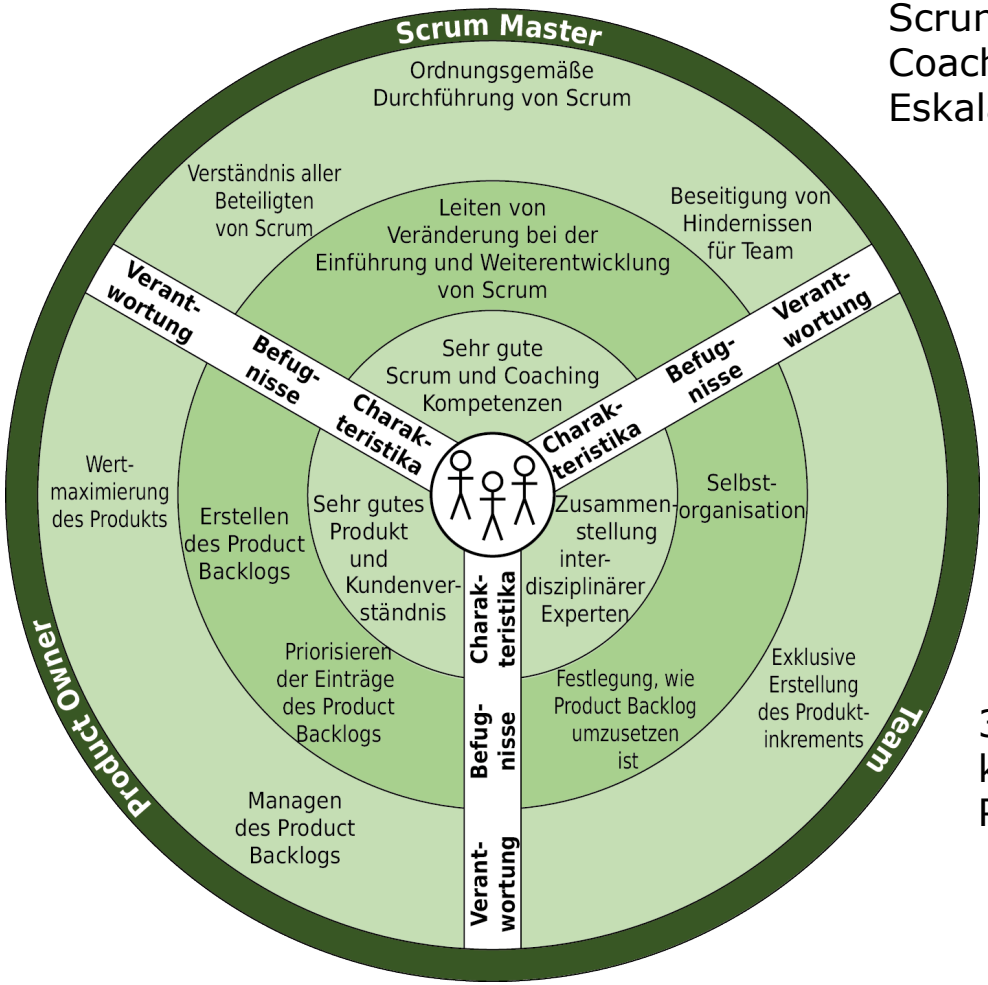
- Wichtige Bestandteile:
  - Scrum Rollen
  - Scrum Aktivitäten (Ereignisse)
  - Scrum Artefakte



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 166

Scrum Rollen

Scrum Experte
Coach
Eskalationsinstanz



Wirtschaftlicher Erfolg und strategischer Beitrag des Produkts

3 bis 8 Personen
keine Hierarchie
Pull-Prinzip

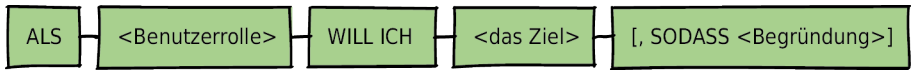
Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 167

Scrum Artefakte

- Artefakte = Zwischen- oder Endergebnisse
- Zielsetzung: Transparenz, Überprüfung und Anpassung im Projektverlauf
- Überblick:
  - Product Backlog
  - Sprint Backlog
  - Produktinkrement
  - Releaseplan
  - Impediment Backlog

Product Backlog

- Priorisierte Sammlung von Anforderungen an das zu entwickelnde Produkt
- Bestandteile:
  - User Story: Anforderung aus Sicht einer bestimmten Rolle

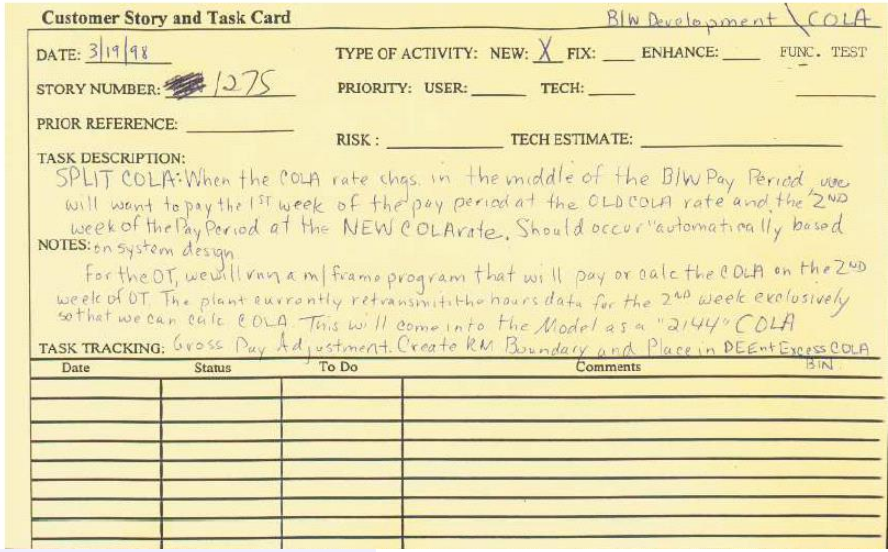


    - sind unabhängig voneinander (**i**ndependent).
    - sind verhandelbar (**n**egotiable).
    - haben Wert für Kunden (**v**aluable).
    - sind hinsichtlich ihrer Größe schätzbar (**e**stimatable).
    - sollen klein sein (**s**mall).
    - sollen testbar sein (**t**estable).

  - Epic: Große, noch vage User Story (Platzhalter für später auszuarbeitende User Stories)

User Story mit Story Points

Story Nr	Story	Abschätzung (Story Points)
1	Als Arzt kann ich alle Patienten sehen, die ich am Tage habe.	3
2	Als Arzt kann ich über die Gesundheitsgeschichte meiner Patienten Auskunft geben.	5
3	Als Assistentin kann ich einem Patienten einen Termin geben.	2
4	Als Assistentin kann ich einem Patienten eine Verschreibung ausdrucken.	1


Customer Story and Task Card

BIW Development \ COLA

DATE: 3/19/98

TYPE OF ACTIVITY: NEW: X FIX: ___ ENHANCE: ___ FUNC. TEST

STORY NUMBER: [redacted] 1275

PRIORITY: USER: _____ TECH: _____

PRIOR REFERENCE: _____

RISK: _____ TECH ESTIMATE: _____

TASK DESCRIPTION:
SPLIT COLA: When the COLA rate chgs. in the middle of the BIW Pay Period user will want to pay the 1ST week of the pay period at the OLD COLA rate and the 2ND week of the Pay Period at the NEW COLA rate. Should occur "automatically based

NOTES: on system design
for the OT, we will run a m/frame program that will pay or calc the COLA on the 2ND week of OT. The plant currently retransmits the hours data for the 2ND week exclusively so that we can calc COLA. This will come into the Model as a "2144" COLA

TASK TRACKING: Gross Pay Adjustment. Create RM Boundary and Place in DEEntExpressCOLA BIN

Date	Status	To Do	Comments

- Funktionalität des Systems wird in Users Stories beschrieben
- User Stories werden auf Story Cards vom Kunden mit höchstens 3 Sätzen notiert
- Kunde ist bei der gesamten Entwicklung dabei! (da Story ggf. nicht detailliert genug)
- Jede Story wird realisiert und in einen Akzeptanztestfall umgesetzt
- Treten bei der Abnahme Fragen auf, gibt es neue Stories und neue Testfälle werden definiert

User Stories beschreiben Teile der zu liefernden Software in „Nutzer-Sprache"

Product Backlogs Using User Stories

User stories describe pieces of software to deliver in the language of someone who will use the software. A **short story title** is written on a card, sticky, or in a list as a **token for conversation**. Stories split into smaller stories and gain more detail over time and through many conversations with contributors in every role.

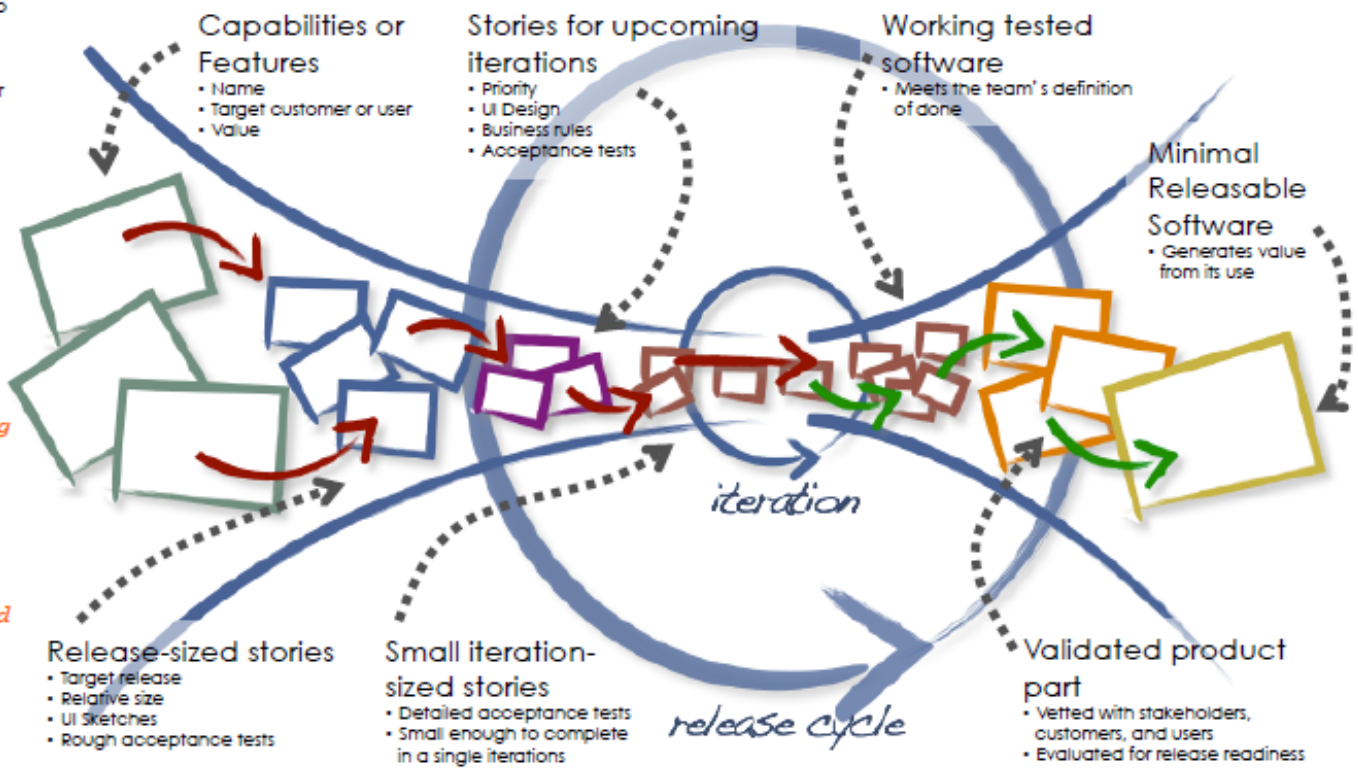
At minimum a story needs:
- Concise title
- Description
- Acceptance criteria

This popular template helps to think through stories from a user and benefits perspective:

as a [type of user]
I want to [perform some action]
so that I can [reach some goal]

Use this template as a thinking tool, not a writing format. Stories without concise titles become hard to organize, discuss, and prioritize.

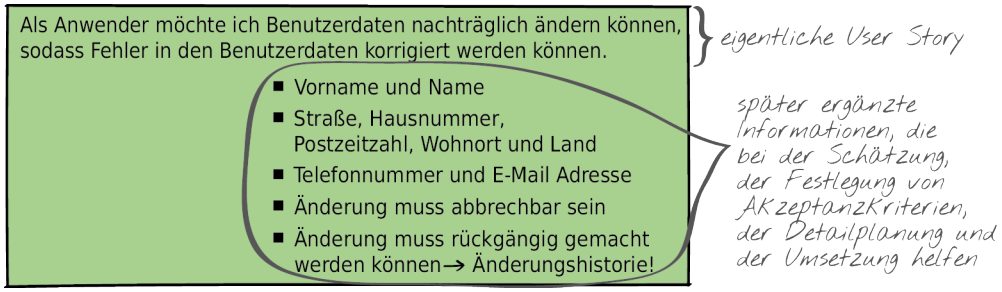
*Think about **who** uses the software, **what** they'll do, and **why**.*



Quelle: Patton, J. (2015): Agile Development & Scrum

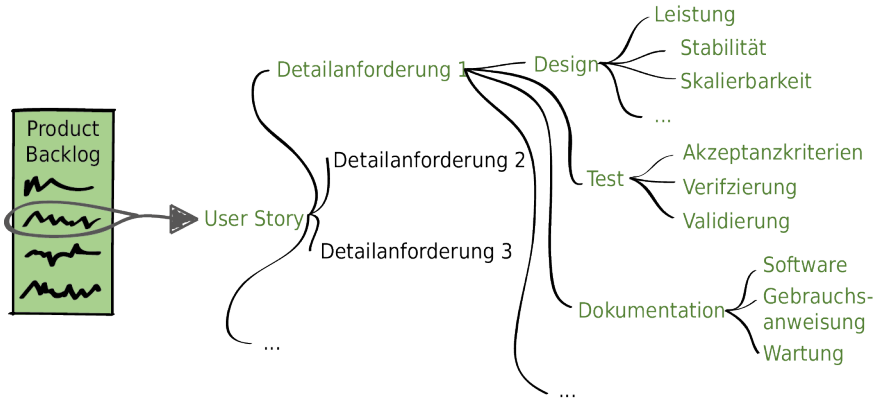
Product Backlog erstellen (1)

User Stories und Epics sammeln (Karten/Post it's – Software)



Benutzerverwaltung

Gemeinsames Verständnis bzgl. vollständiger Umsetzung der User Story erarbeiten (Definition of Done)



Product Backlog erstellen (2)

- Größe der User Stories schätzen
  (abhängig von der Größe – Komplexität der User Story => Einheit: Story Points)
- User Story anhand Geschäftswert priorisieren
  (Sortierung von wichtig zu unwichtig; Punktevergabe; Klassifizierung)
- User Stories organisieren und verwalten
  (bspw. Pinnwand, elektronische Pinnwand oder in Listen)

	A	B	C	D	E	F
1	**ID**	**User Story**	**Größe [Story Points]**	**Priorität**	**Ergänzende Notizen**	
2	A5	Als Anwender will ich ...	13	100		
3	M3	Als Manager will ich ...	20	115		
4	H2	Als Hersteller will ich ...	8	150	siehe auch Dok. D13	
5	A2	Als Anwender will ich ...	100	200		
6	A1	Als Anwender will ich ...	3	205		
7	H1	Als Hersteller will ich ...	40	210		
8	M2	Als Manager will ich ...	8	213		
9	A4	Als Anwender will ich ...	5	250	mit Produkt T20 abstimmen	
10	M1	Als Manager will ich ...	1	260		
11	S1	Als Servicetechniker will ich ...	40	270		
12	A3	Als Anwender will ich ...	20	280	Bestätigung fehlt noch	
13						

Sprint Backlog

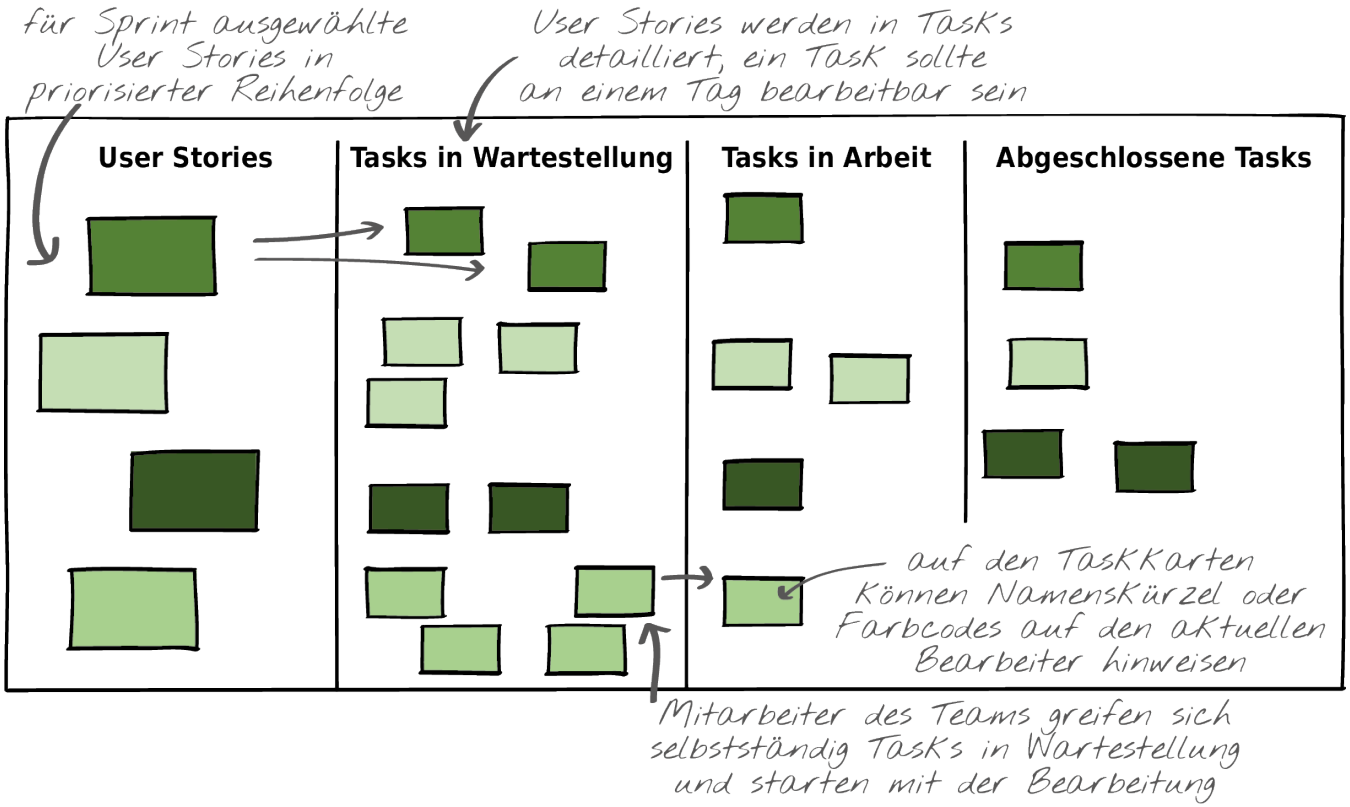
- Sammlung, der im aktuellen Sprint anstehenden Aufgaben, die aus den User Stories abgeleitet wurden
- entsteht im Rahmen der Sprint Planung

  (1) Auswahl der User Stories, die im folgenden Sprint umzusetzen sind, aus dem Product Backlog (Berücksichtigung von Team-Velocity)
  - Anzahl an Story Points, die ein Team in einem Sprint erledigen kann
  - Wichtige Kennzahl zur Leistungsfähigkeit eines Teams

  (2) Ableitung von Aufgaben (Tasks) und Verteilung der Story Points => jede Task hat eine Größenschätzung

Sprint Backlog visualisieren (1)

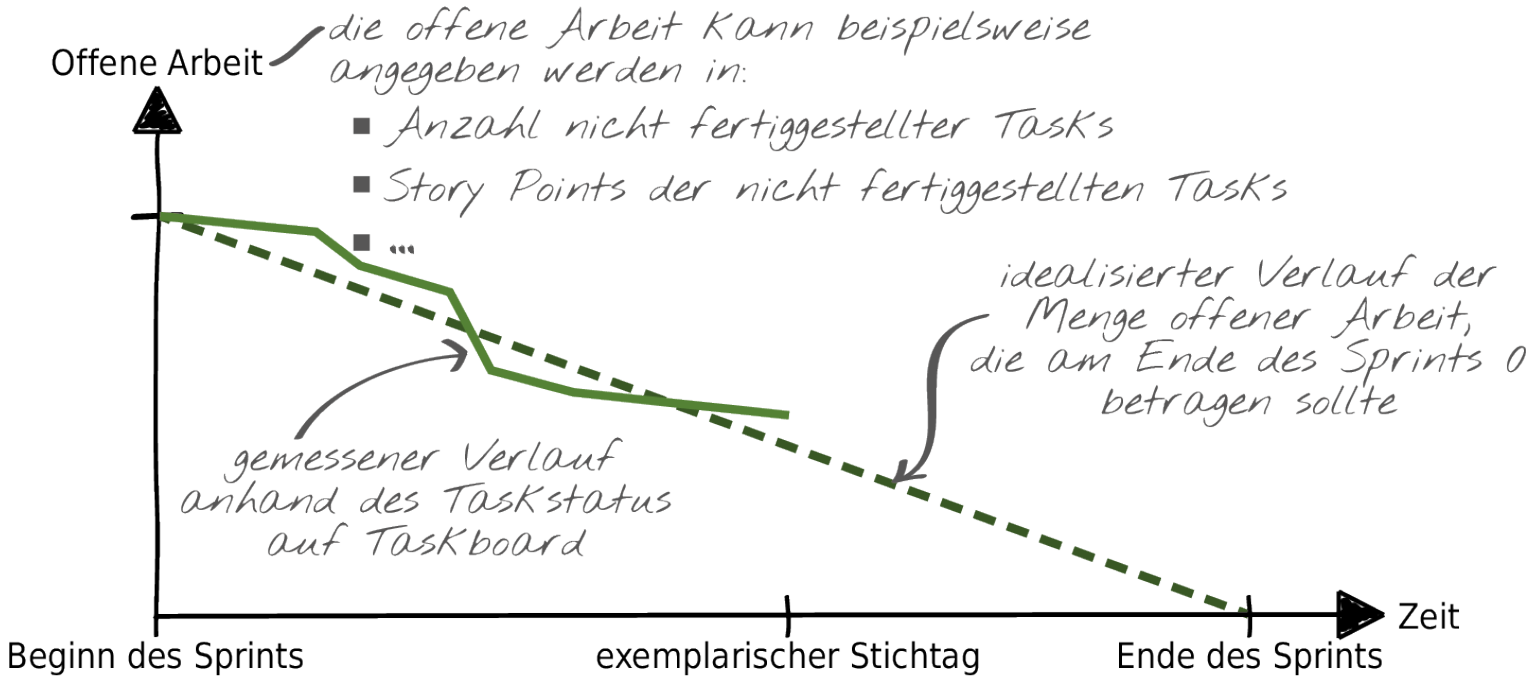
- Taskboard



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 175

Sprint Backlog visualisieren (2)

- Burndown Chart



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 176

Produktinkrement

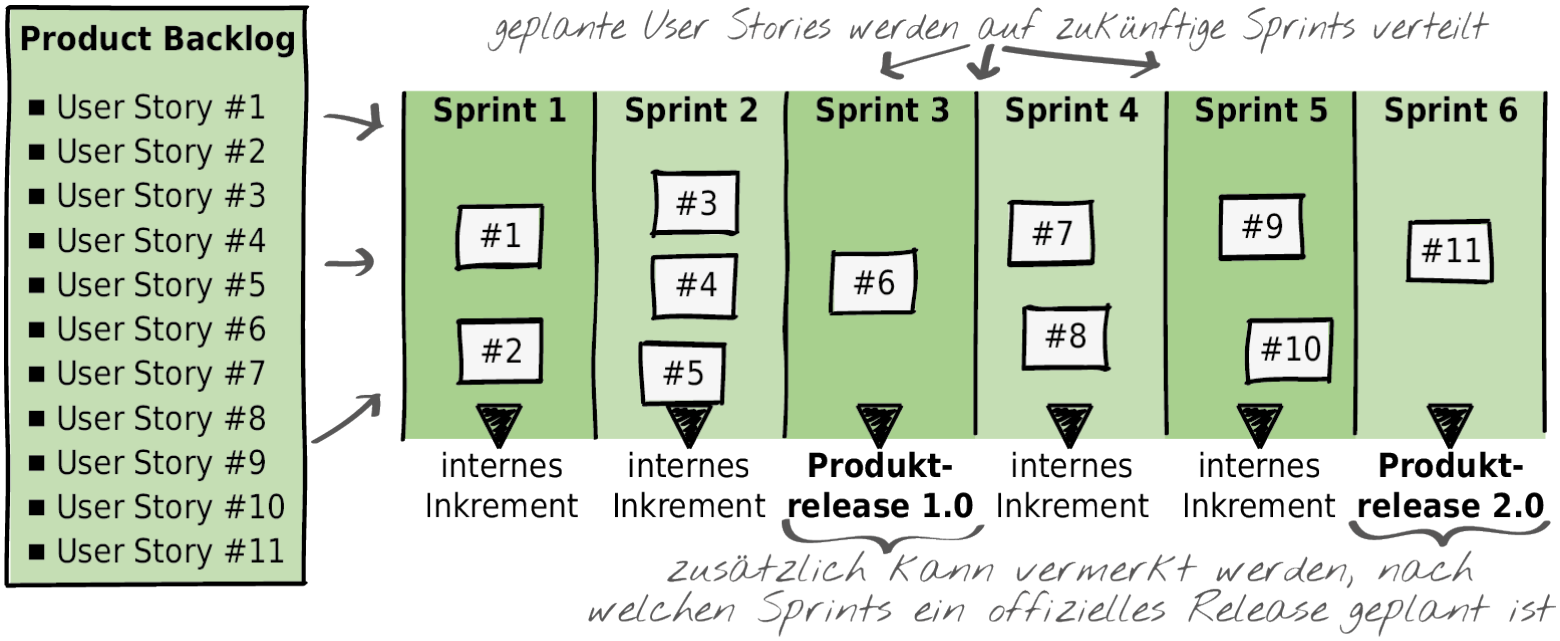
- (Teil-)Ergebnis eines Sprints
- Beispiele:
  - neue Software
  - neue Hardware
  - neues System oder Endprodukt
  - Konzept für eine Prozessverbesserung oder eine Veranstaltung.

Jedes einzelne Produktinkrement ist lauffähig bzw. vorzeigbar!

- Product Owner oder Kunde müssen ausprobieren können
- Rückmeldungen einholen
- Epics und User Stories konkretisieren
- Gewonnenen Erfahrungen fließen in die nächste Sprint Planung

Releaseplan

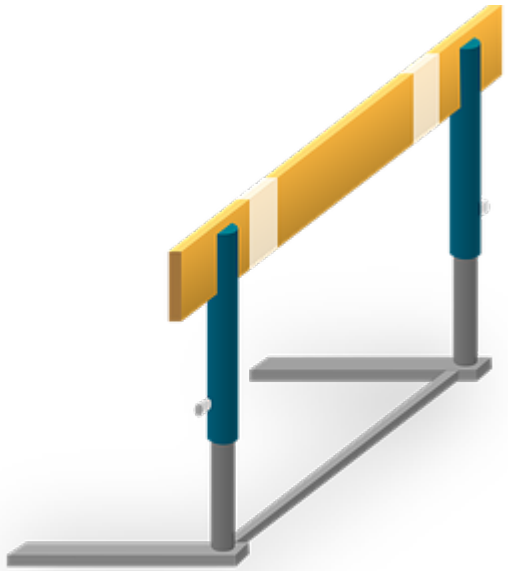
- Übersichtsplan, aus dem ersichtlich ist, in welchem Sprint welche User Stories umgesetzt werden sollen



Quelle: Tminger H. (2017): Modernes Projektmanagement, S. 178

Impediment Backlog

- Auflistung aller Hindernisse, die dem Team bei der Arbeit begegnen
  - Von Scrum Master geführt
  - Bei täglichen Teambesprechungen (Daily Scrums) gefüllt

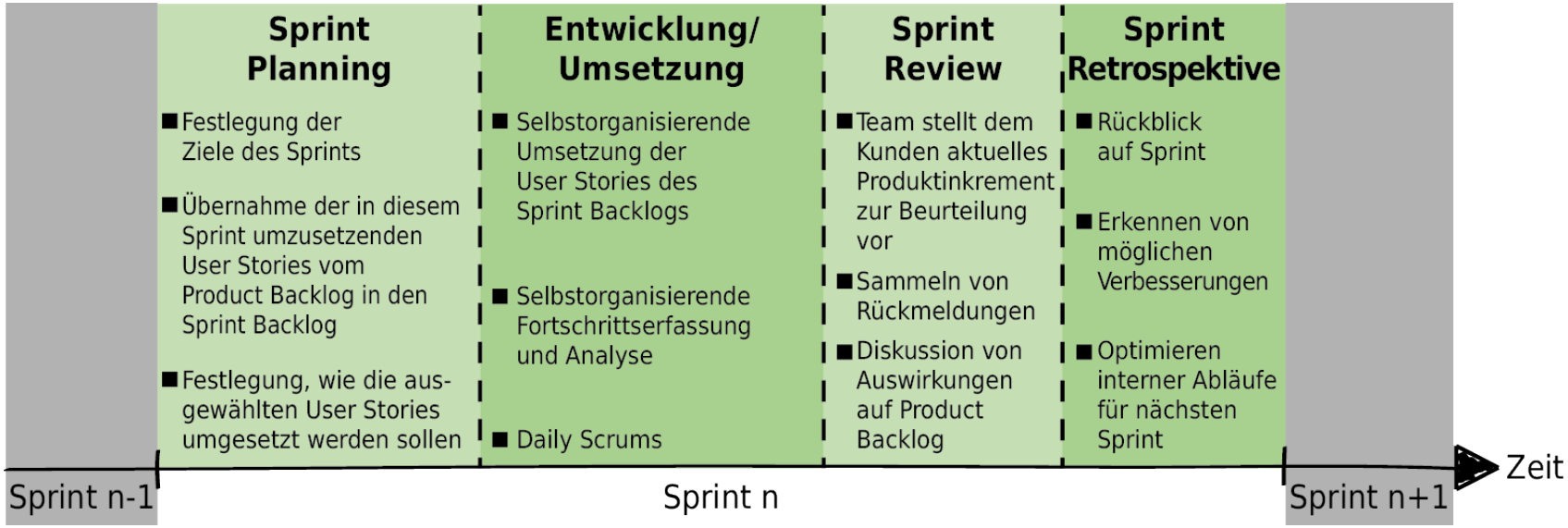


Scrum Aktivitäten

- Beschreibung der einzelnen Schritte in Scrum
  - Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospektive

Sprint

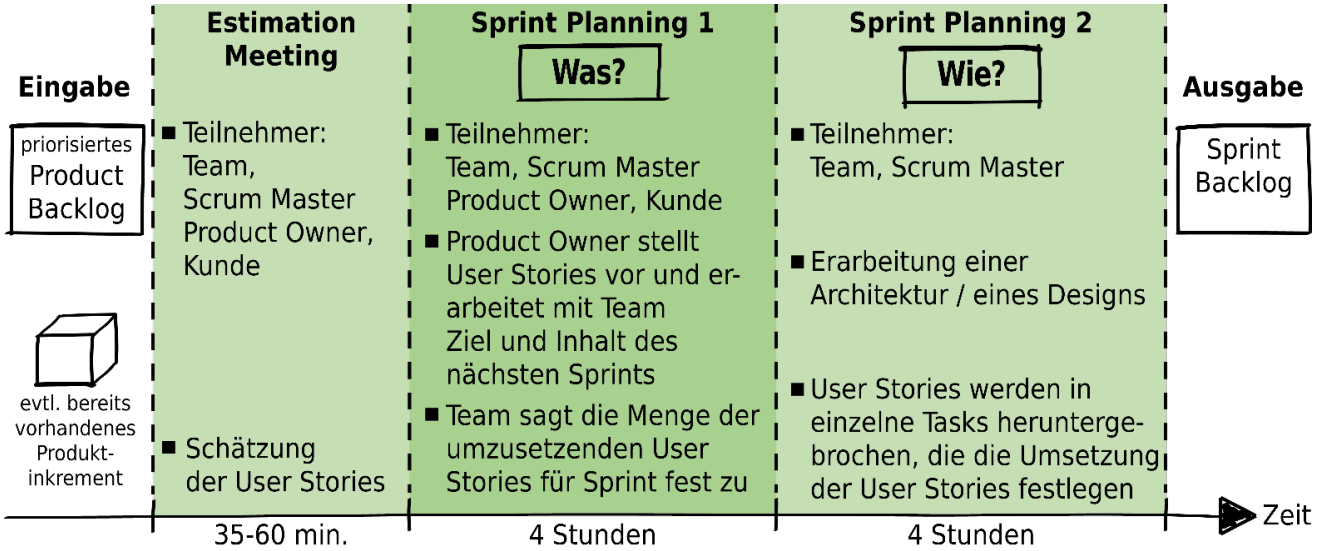
- Fester Zeitraum, in dem die Anforderungen an das Produkt umgesetzt werden
- Verschiedene einzelne Schritte innerhalb eines Sprints, die sich in allen Sprints wiederholen



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 179

Sprint Planning

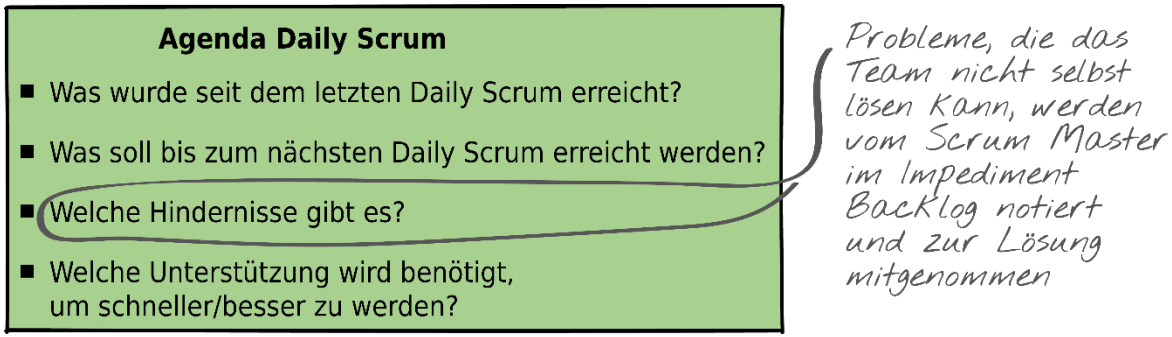
- Beantwortung folgender zwei Fragen:
  - Was ist das Ziel des folgenden Sprints und welche User Stories sollen zur Zielerreichung im kommenden Sprint umgesetzt werden?
  - Wie erfolgt die Umsetzung der ausgewählten Anforderungen?



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 181

Daily Scrum

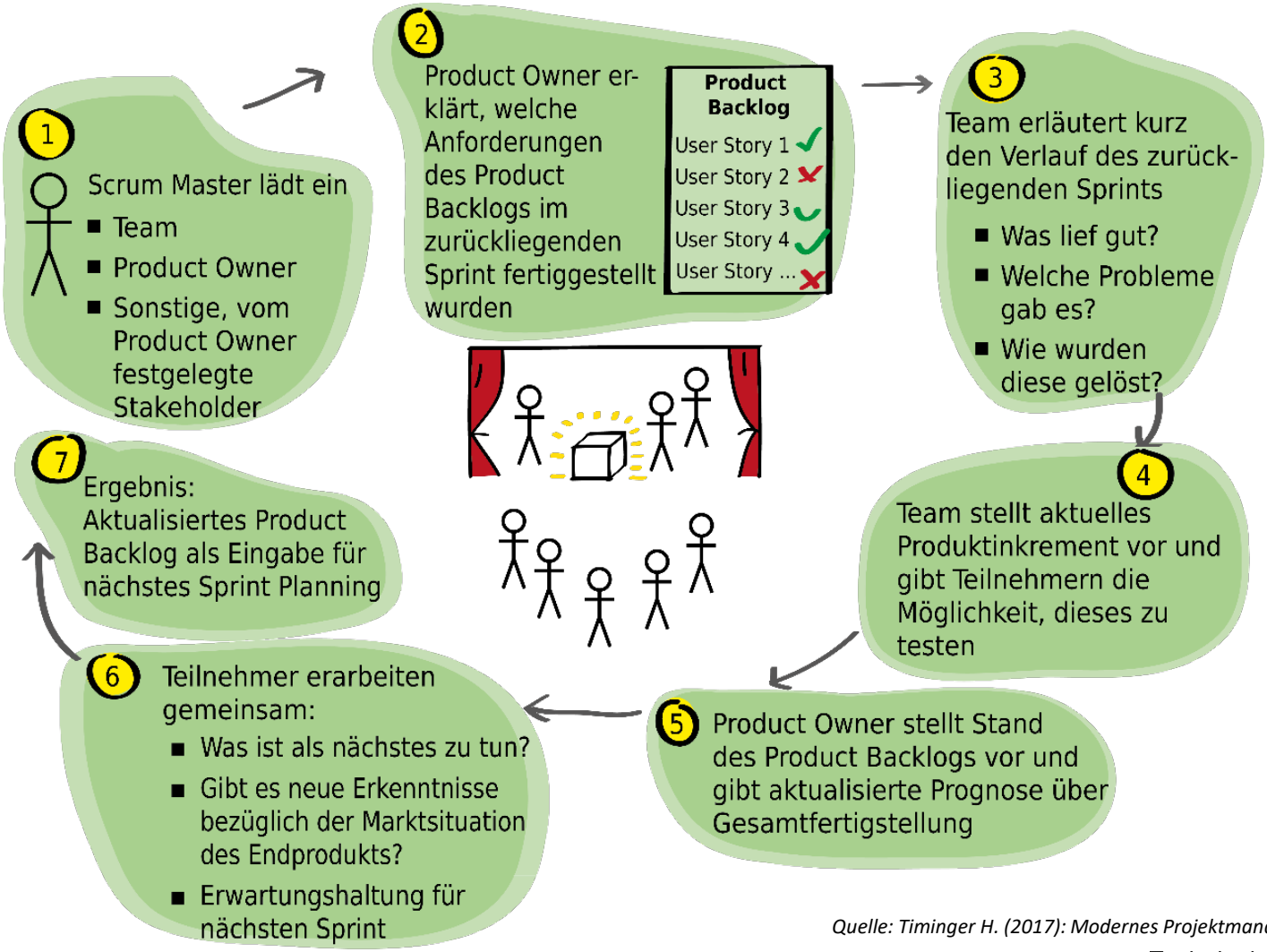
- Kurze Besprechung zur
  - Statuserfassung und –analyse
  - Identifikation von Hindernissen
  - Synchronisation der Aufgaben im Team
  - Einleiten von Steuerungsmaßnahmen
  - Planung der nächsten 24 Stunden
- Typische Agenda:



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 184

Sprint Review

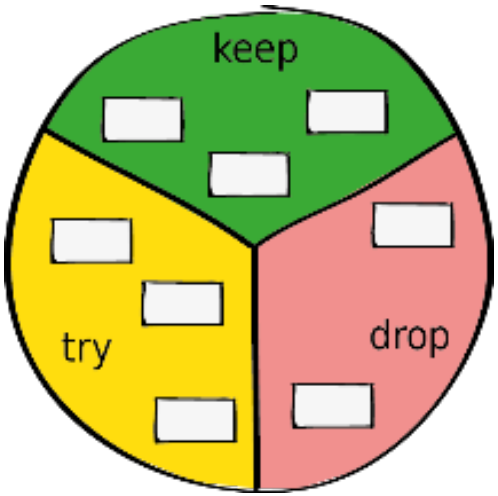
Überprüfung des Produktinkrements



Quelle: Timinger H. (2017): Modernes Projektmanagement, S. 184

Sprint Retrospektive

- Analyse der teaminternen Zusammenarbeit und Schnittstellen zu anderen Bereichen des Unternehmens



- Möglicher Ablauf:
  - Brainstorming im Team zu positiven und negativen Aspekten des zurückliegenden Sprints sowie Ideen für zukünftige Verbesserungen
  - Sammlung und Kategorisierung im „keep-drop-try-Schema"
  - Bewertung und Diskussion mit unterschiedlichen Sichtweisen
  - Ableitung und Priorisierung von konkreten Maßnahmen zur Veränderung
  - Für wichtige Maßnahmen Verantwortlichen benennen und Umsetzung besprechen