

Themenccluster: Agile und schlanke Methoden

Thema: SCRUM

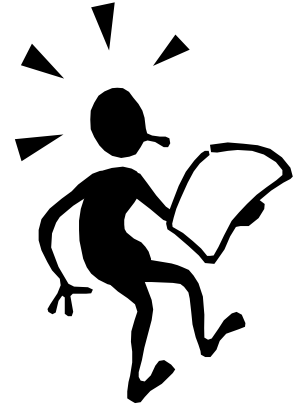
Dr. Walter Rafeiner-Magor

08.05.2011

SCRUM

● Was ist Scrum?

- Ein Begriff aus dem Rugby-Sport (Übersetzung: „Gedränge“)
- Es handelt sich hierbei um einen speziellen Spielzug, welcher genau einstudiert werden muss (um erfolgreich zu sein).
- agil und schlank



Voraussetzung
für den Erfolg sind
disziplinierte Teams!

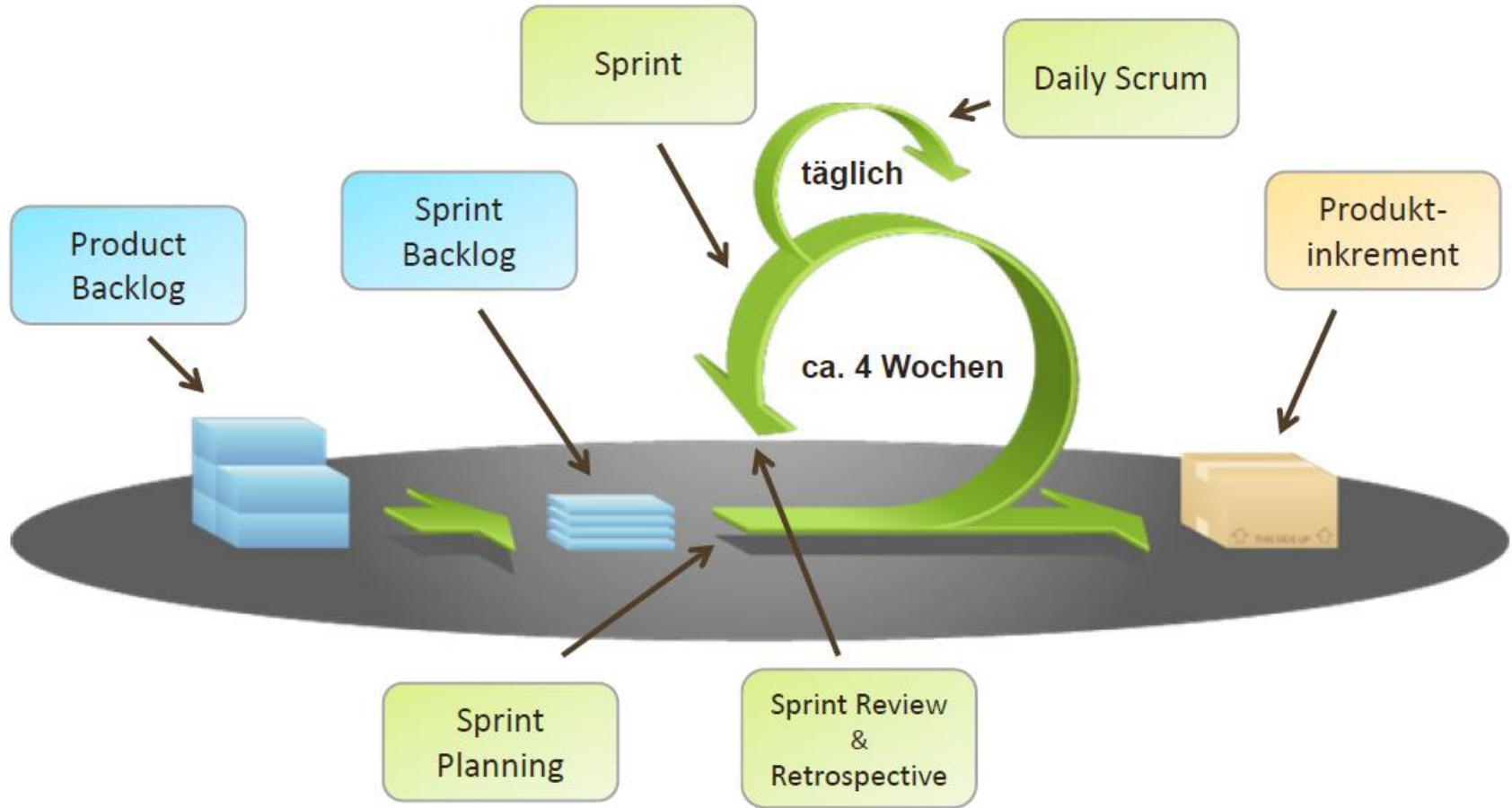
SCRUM

● Was ist Scrum?

- **1986:** Nonaka und Takeuchi beschreiben, dass kleine hochvernetzte und interdisziplinäre Teams die besten Resultate erzielen und bezeichnen dieses Vorgehen als Scrum.
- **1990:** DeGrace und Stahl erwähnen erstmals Scrum im Zusammenhang mit Software
- **1993:** Jeff Sutherland setzt Scrum bei Easel Corp. ein
- **1996:** Ken Schwaber liefert bei der OOPSLA 96 gemeinsam mit Jeff Sutherland eine erste Definition von Scrum
- **2001:** Als agiles Framework verkörpert Scrum die Werte des agilen Manifests <http://agilemanifesto.org/iso/de/>



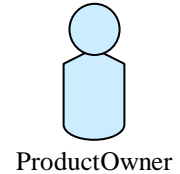
Der Scrum-Prozess



Rollen und Verantwortlichkeiten



Rollen in Scrum: ProductOwner



● **Kernaufgaben:**

- Anforderungsmanagement
- Zusammenarbeit mit dem Team
- Stakeholdermanagement

● **Umfang:**

- Meist Vollzeitaufgabe

● **Fähigkeiten:**

- Produkt- oder Marketingmanager

● **Wichtige Fragen an die Rolle:**

- Wer übernimmt die Verantwortung für den Erfolg des Produktes?
- Wer hat die Kompetenz und die Macht, über die Gestaltung des Produktes zu entscheiden?
- Wann ist das Produkt ein Erfolg? Gibt es Stufen zum Erfolg?
- Gibt es eine nachvollziehbare Kosten-/Nutzenbetrachtung?
- Wer vermittelt eine Produktvision, die die besten Mitarbeiter motiviert?

Rollen in Scrum: Scrum-Team



- **Kernaufgaben:**

- Sämtliche Arbeiten die zur Erreichung eines Sprint-Ziels erforderlich sind

- **Größe des Teams:**

- 5-7 Vollzeitbeschäftigte

- **Fähigkeiten:**

- Alle Fachbereiche, die zur Fertigstellung des Produktes benötigt werden

- **Besonderheit:**

- Scrum-Teams organisieren sich selbst

- **Wichtige Fragen an die Rolle:**

- Ist das Team ein Team?
- Kennt das Team die Produktvision und steht hinter der Vision?
- Sind Aufgaben und Zusammenhänge inhaltlich und zeitlich für jeden klar?
- Übernimmt das Team die Verantwortung für die Erstellung der Lösung? Hat das Team die Aufwände geschätzt?
- Hat das Team die Kompetenz und die Mittel, um eine Lösung zu erstellen?

Rollen in Scrum: ScrumMaster



ScrumMaster

● **Kernaufgaben:**

- Enge Zusammenarbeit mit dem Team
- Beseitigung von Hindernissen, die das Team von effizienter Arbeit abhalten
- Schulung und Überwachung des Scrum-Prozesses

● **Umfang:**

- Meist Vollzeitaufgabe (Der Umfang ändert sich während des Projektes)

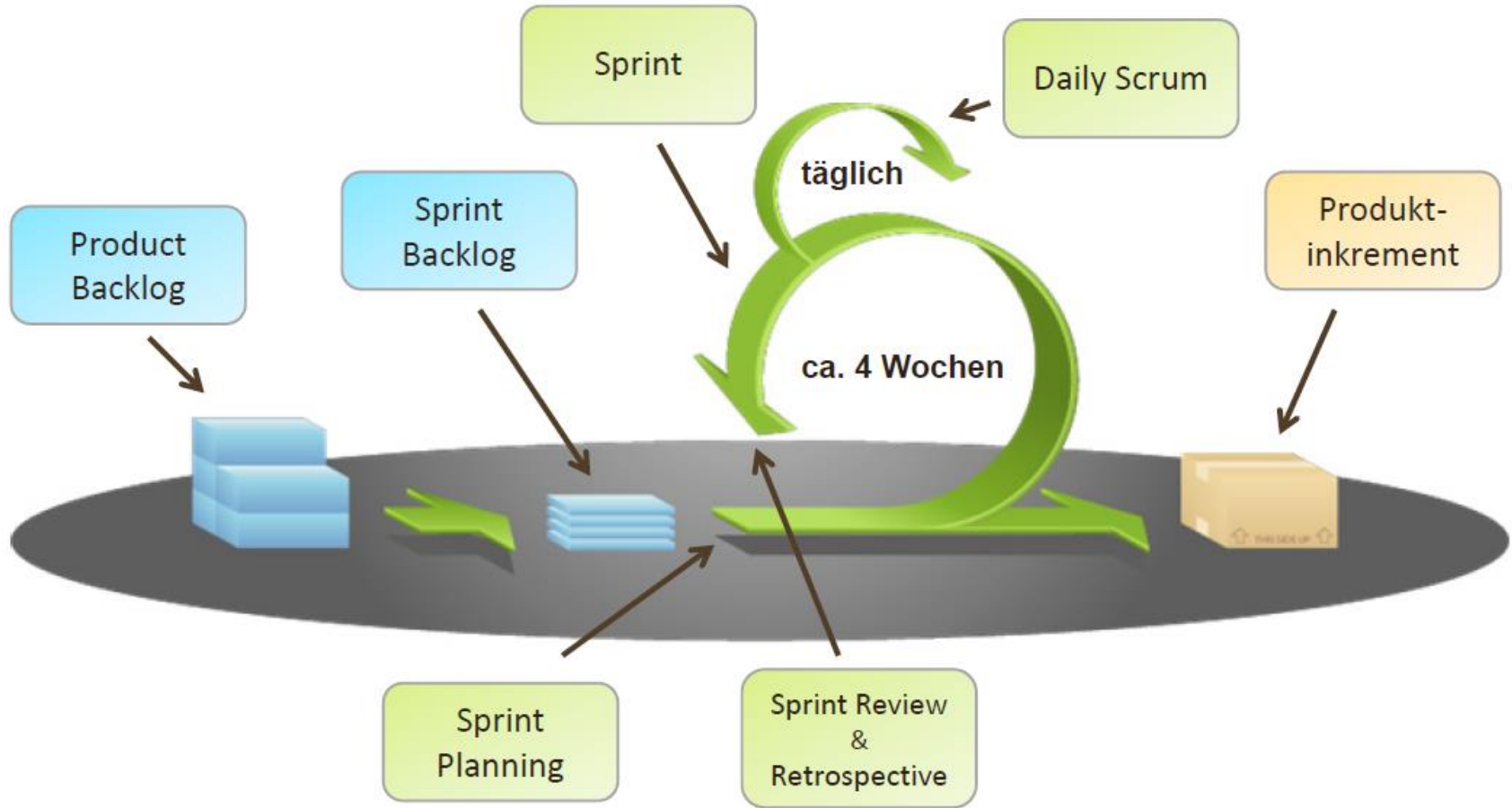
● **Leitsatz:**

- „Dienen statt Führen“

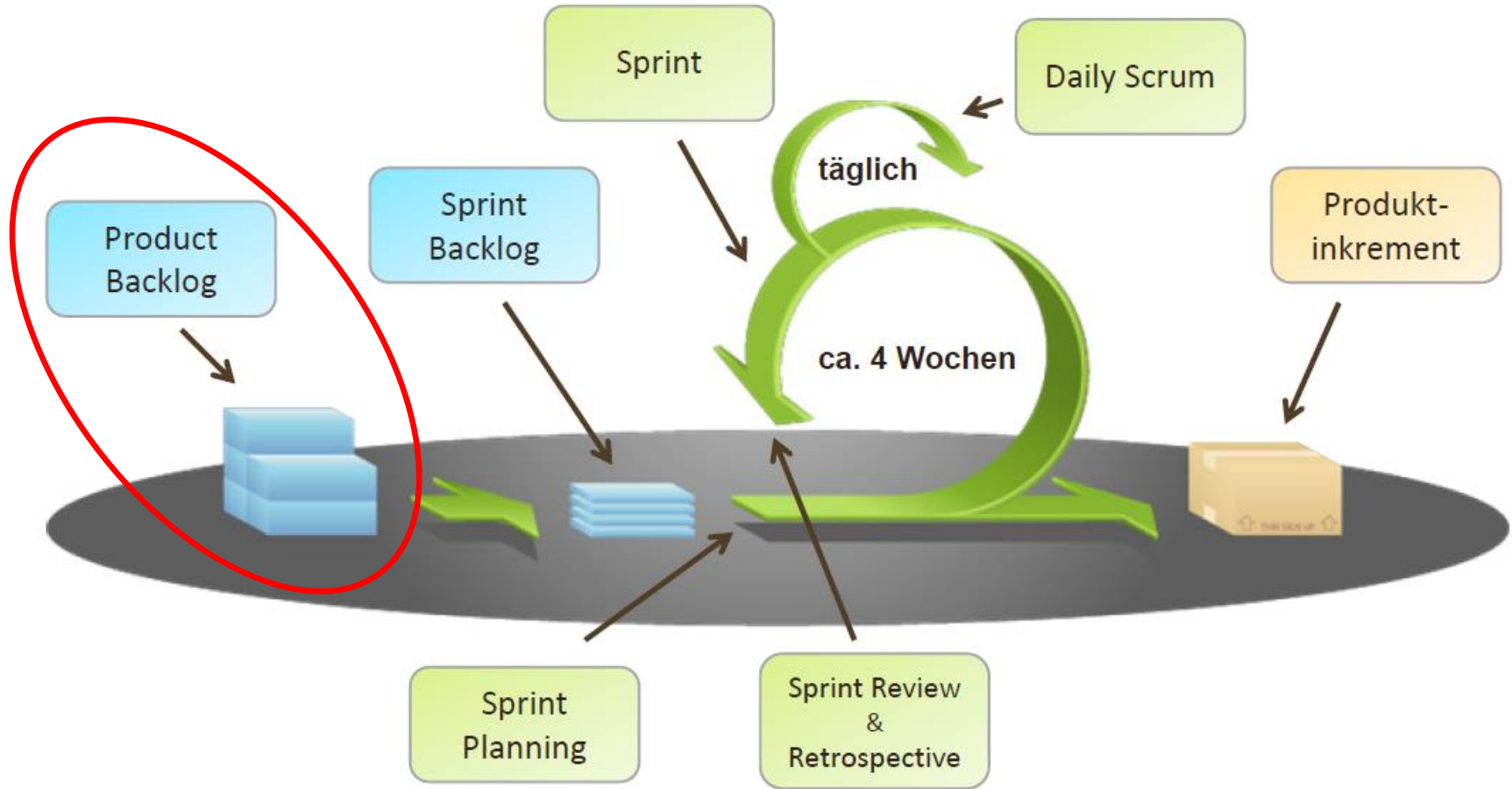
● **Wichtige Fragen an die Rolle:**

- Wer moderiert Kommunikations- und Teamprozesse?
- Werden relevante Informationen erhoben und allen Beteiligten zugänglich gemacht?
- Werden Entscheidungen kooperativ und effizient herbeigeführt?
- Wird bewusst auf Wandel reagiert?
- Werden Schwachstellen konsequent beseitigt?
- Werden Timeboxes eingehalten?

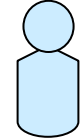
Der Scrum-Prozess



Product Vision ➡ Product Backlog



Product Vision



ProductOwner

● **Inhalt:**

- Idee eines neuen Produkts, das alle Beteiligten überzeugt!

● **Ziel:**

- Darstellung des Nutzens (eigenes Unternehmen, Investor oder Auftraggeber)

● **Verantwortung:**

- Die Kommunikation der Zielsetzung sowie Ableitung von Entscheidungen (ProductOwner)

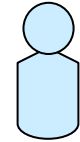
● **Entstehung abhängig von der Projektart:**

- Auftragsentwicklung: Erarbeitung zusammen mit dem Kunden
- Eigenentwicklung: Erarbeitung meist zusammen mit dem Management
- Weiterentwicklung: Entstehung durch den Nutzen, der aus einer verbesserten Version entsteht

● **Die Product Vision schafft Identität für alle Beteiligten:**

- Wer nutzt das Produkt?
- Was sind die wesentlichen Bedürfnisse der Kunden?
- Wie wird der Erfolg gemessen? Gibt es Stufen zum Erfolg?
- Welcher Zieltermin besteht? Budget?

Strategische Releaseplanung



ProductOwner

● Ziel:

- Aufteilung des Projektes in sinnvolle Teile

● Inhalt:

- Anzahl der Sprints
- Kapazität des Teams (Velocity)
- Anforderungen

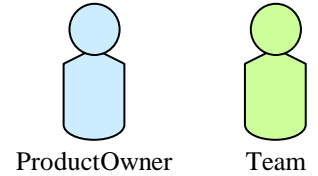
● Verantwortung:

- Die Verantwortung der Umsetzung liegt beim ProductOwner

● Wichtige Punkte der Umsetzung:

- Wenig (überschaubare) Funktionalität in einer Release verankern.
- Keine Big-Bang-Releases.
- Jede Release liefert eines oder mehrere fertige Features.
- Prinzip der inkrementellen Innovation.

Product Backlog (1/2)



● Ziel:

- Anforderung, Funktionalitäten und deren Priorisierung

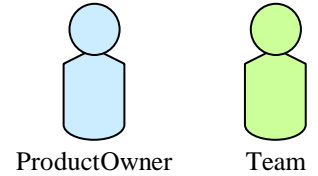
● Verantwortung:

- Die Erstellung und Verwaltung liegt in der Zuständigkeit des ProductOwners (ggf. unter Zuhilfenahme des Teams).

● Umsetzung:

- Die Anforderungen/Funktionalitäten werden im zentralen Product Backlog gesammelt und regelmäßig aktualisiert.
- Zu Beginn des Projekts sind BacklogItems nur grobgranular und unvollständig beschrieben
- Im Laufe des Projekts werden diese präzisiert und vervollständigt.
- Keinen zusätzlichen Change-Request-Prozess.
- Beschreibung nicht-funktionale Anforderungen
- Benutzerschnittstellen, Testumgebungen und Fehlerbehebungen
- Jede Anforderung enthält Angaben zu:
 - Priorität, Beschreibung (User Story), Akzeptanzkriterien, Risiken und Aufwandschätzung
- Das Product Backlog enthält keine Aufgaben oder Aktivitäten (diese sind in den Sprint Backlogs enthalten)!

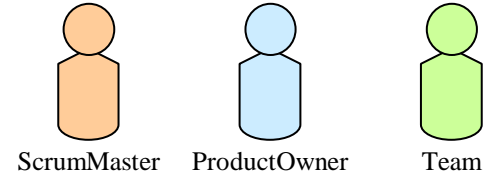
Product Backlog (2/2)



● Möglicher Aufbau:

Prio	Themengruppe	Item-Beschreibung (User Story)	Akzeptanzkriterien Risiken	Aufwand [Story-Points]
1	Online-Reservierung	Kunde benötigt Login	Erfolgreiche Anmeldung	2
2	Online-Reservierung	Kunde will Auto-Kategorie auswählen	Kategorien werden aufgelistet, Kategorie kann ausgewählt werden	5
3	Online-Reservierung	...		

Exkurs: User Story



● Inhalt:

- Die wesentlichen Punkte (ProductVision) sind Ausgangspunkt für User Stories

● Kriterien: INVEST

- Independent: Stories können so unabhängig voneinander in eine Auslieferungsreihenfolge gebracht werden.
- Negotiable: Eine Story ist kein Vertrag, sondern ein Versprechen zur Diskussion.
- Valuable: Die Story soll Geschäftswert schaffen.
- Estimable: Die Story soll in ihre Größe durch das Team abschätzbar sein.
- Small: Die Story soll in einem Sprint auslieferbar sein.
- Testable: An jedem Sprint Abschluss müssen Testergebnisse sichtbar sein.

● Wichtige Fragen bei der Erstellung:

- Wer? Was? Warum?
- Versprechen zur Diskussion!
- Kein Use Case!

Exkurs: User Story

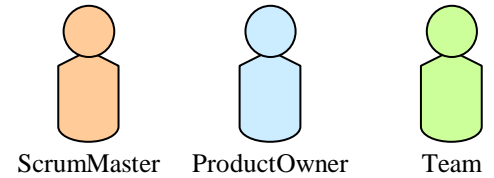
● Möglicher Aufbau:

<div>Rolle Wer?</div> <div>Id:</div>	<div>Größenschätzung</div> <div>Aufwand [Story-Points]</div>	Akzeptanzkriterien, Risiken
<div>Als:</div>		
<div>Möchte ich:</div>		
<div>Weil:</div>		
<div>Wert, Nutzen Warum?</div>	<div>Feature Was?</div>	<div>Ausgangspunkt für Akzeptanztests</div>

Walter Rafeiner-Magor

16

Exkurs: User Story



● Aufteilungsmöglichkeiten:

- User Stories, die einen Sprint sprengen würden, sollten verkleinert oder geteilt werden

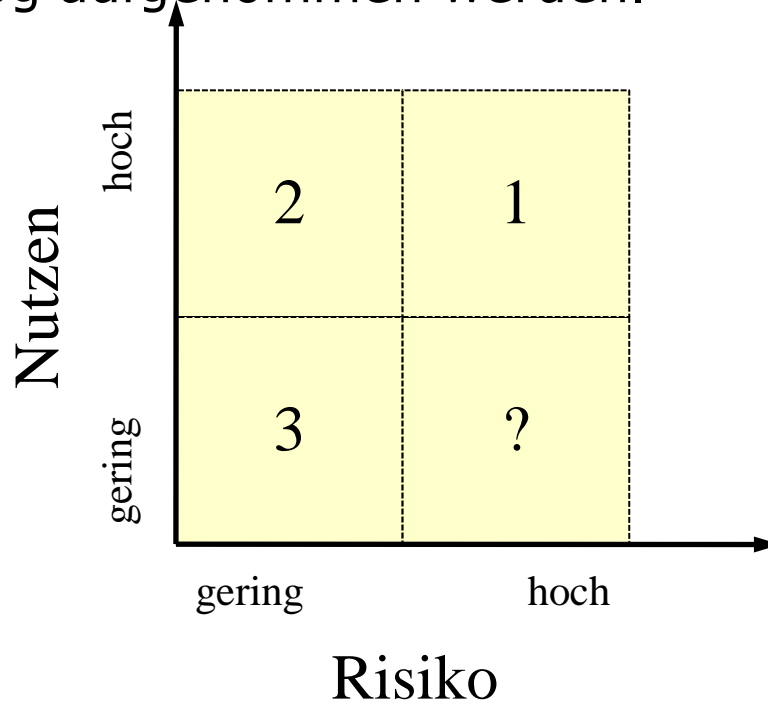
● Kriterien:

- **Arbeitsfolge:** Stories werden in die Folge der zu erledigenden Arbeitsschritte aufgeteilt, z.B. nach aufeinanderfolgenden Bildschirmen.
- **Operation:** Stories werden nach Create, Read, Update und Delete (CRUD) Operationen aufgeteilt.
- **Qualität:** Stories werden aufgeteilt in funktionale und nicht-funktionale (z.B. Performance, Usability) Bestandteile.
- **Rolle:** Stories werden aufgeteilt nach den Benutzerrollen, die durch die Story betroffen sind.
- **Daten:** User Stories werden nach Variationen der zu verarbeitenden Daten aufgeteilt.
- **Implementierung:** User Stories werden nach der Implementierungskomponente aufgeteilt, z.B. Eingabe, Verarbeitung, Ausgabe und Speicherung.

Exkurs: User Story

● **Priorisierung:**

- Stories mit höchster Priorität werden als erste durch das Team ausgeliefert.
- Neu entstehende Stories können jederzeit in den Product Backlog aufgenommen werden.



Exkurs: User Story

"Prediction is very difficult,
especially about the future."

Niels Bohr



Team

● Aufwandsschätzung:

- Die Schätzung erfolgt meist in Estimation-Meetings durch das Team.

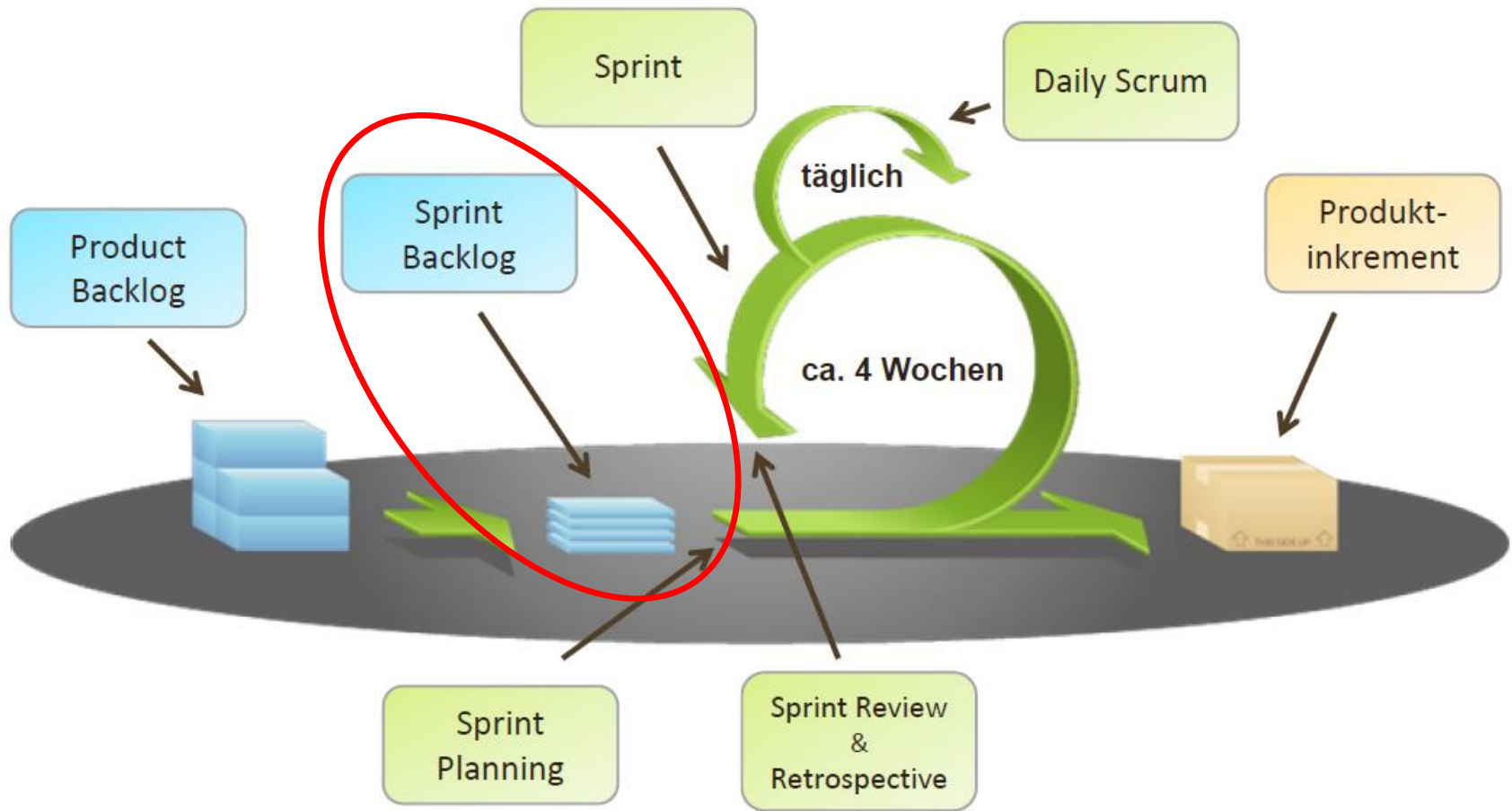
● Velocity:

- Anzahl der Story Points, die das Team innerhalb einer User Story abarbeiten kann

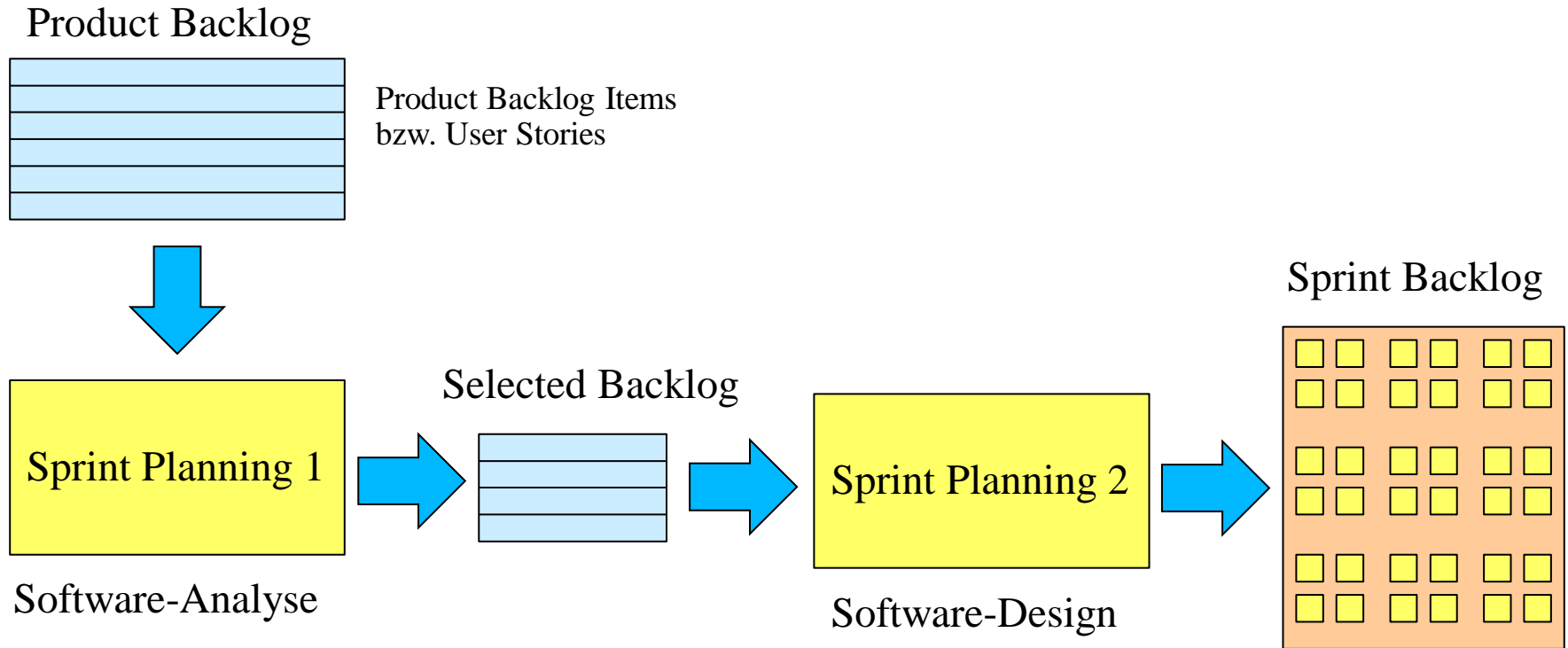
● Umsetzung:

- Als Maßeinheit wird nicht in Stunden oder Tagen gerechnet, sondern in relativen „Story Points“.
Die Aktivität mit dem geringsten Aufwand erhält "1", eine mit doppeltem Aufwand eine "2" etc.
Dies erleichtert die Aufwandschätzungen, weil am Anfang die "Entwicklungsgeschwindigkeit" des Teams noch nicht feststeht.
- Eine Schätzung ist KEIN Commitment!
- Schätzungen finden Eingang in den Product Backlog und den Release Plan.

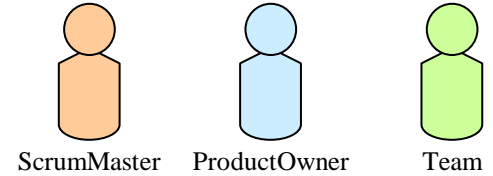
Product Backlog ➡ Sprint Backlog



Sprint Planning



Sprint Planning Meeting



● **Vorbereitung:**

- Organisation von Termin & Räumlichkeiten
- Identifizierung des Sprint Goals
- Vorauswahl passender Anforderungen
- Gegebenenfalls Detaillierung vorhandener Anforderungen

● **Inhalt des Meetings (Teil 1):**

- Team soll Verständnis über anstehende Aufgaben erhalten
- Auswahl der zu bearbeitenden Anforderungen im nächsten Sprint
- Verpflichtungserklärung des Teams (Commitment)

● **Inhalt des Meetings (Teil 2):**

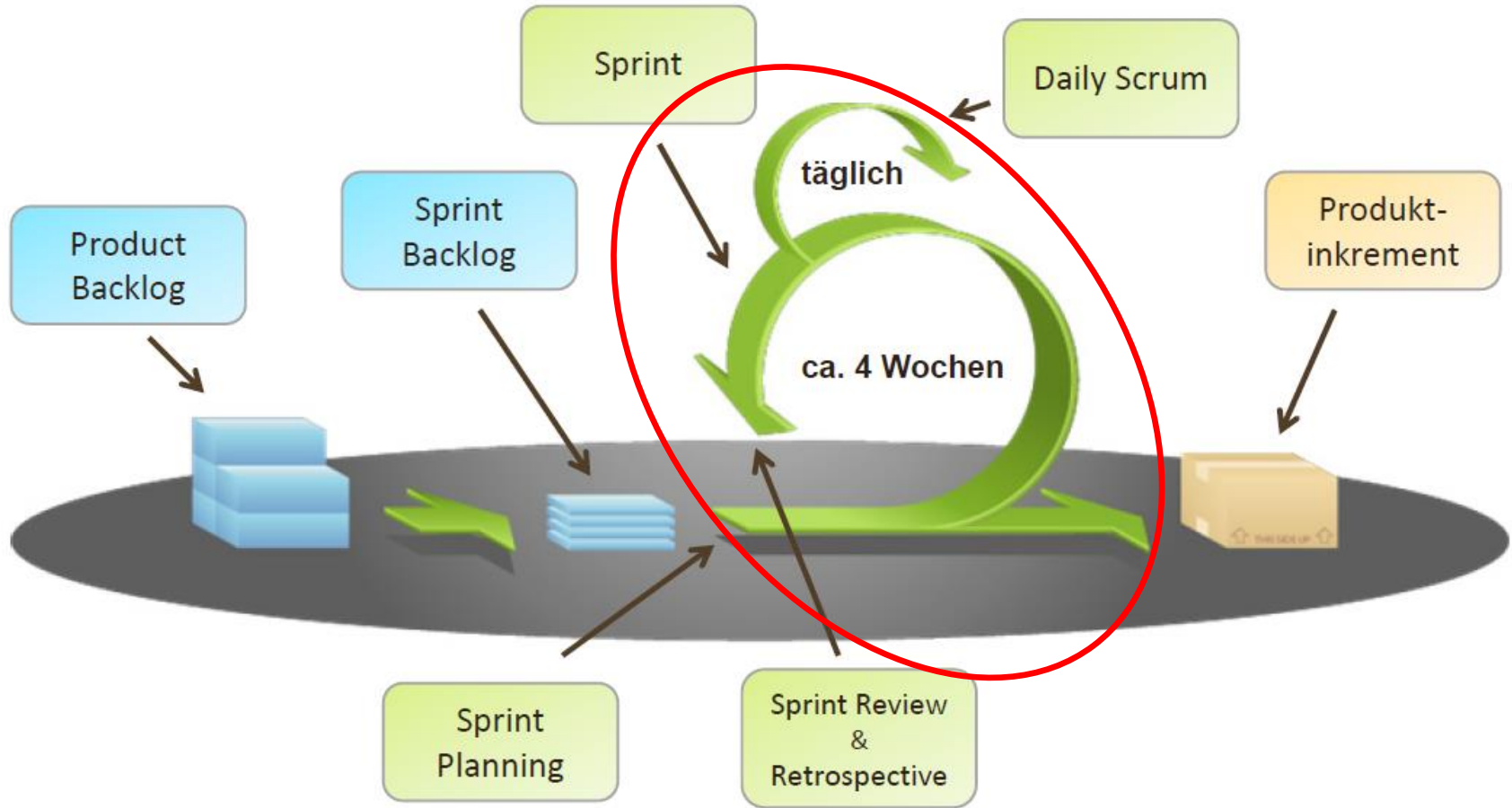
- Ermittlung aller zur Umsetzung erforderlichen Aktivitäten
- Erste Diskussion über Architektur, Design, Konventionen etc.
- Erstellung des Sprint Backlogs

Sprint Backlog

● Möglicher Aufbau:

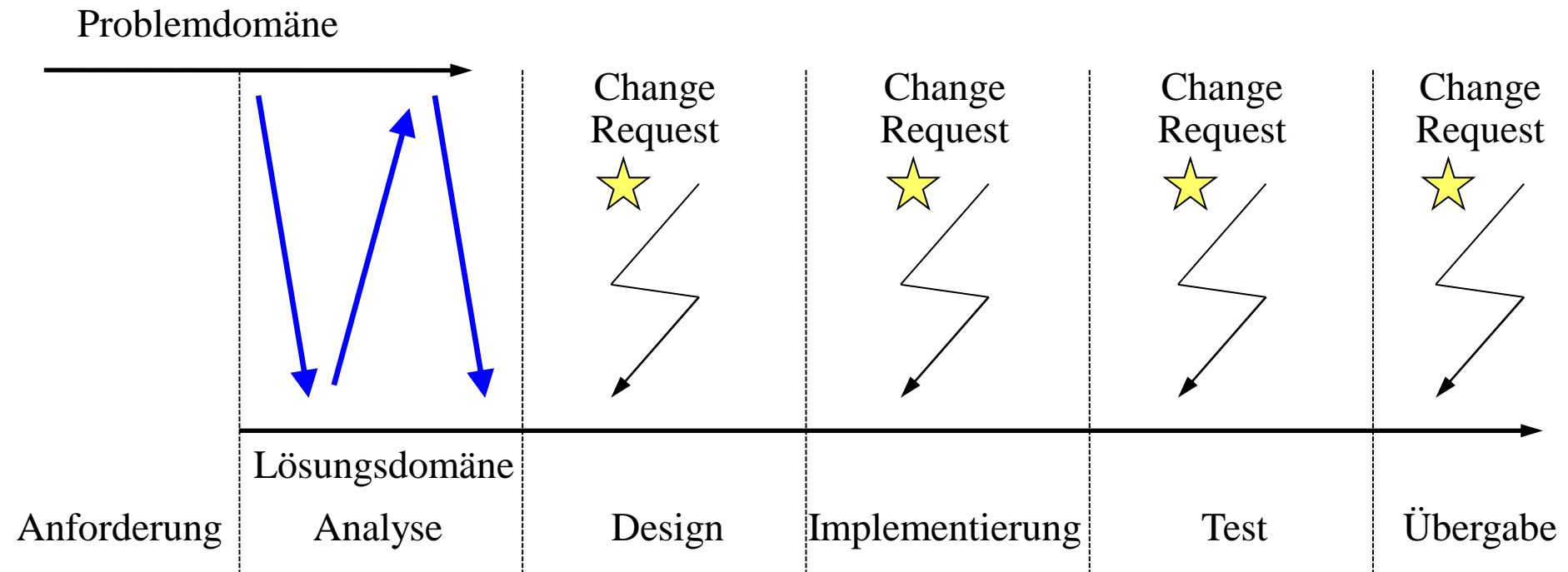
Prio	User Story	Offen	In Arbeit (Wer)	erledigt	Teststatus
1	US A	Aktivität A4	A2 (Fritz)	A1	A1 ok
		Aktivität A5	A3 (Franz)		
2	US B	Aktivität B2	B1 (Paul)		
		Aktivität B3			
3	US C		
			

Der zentrale SCRUM-Prozess



Exkurs: Timeboxing

● Softwareentwicklung als definierter Prozess.



Interdomänen-Kommunikation



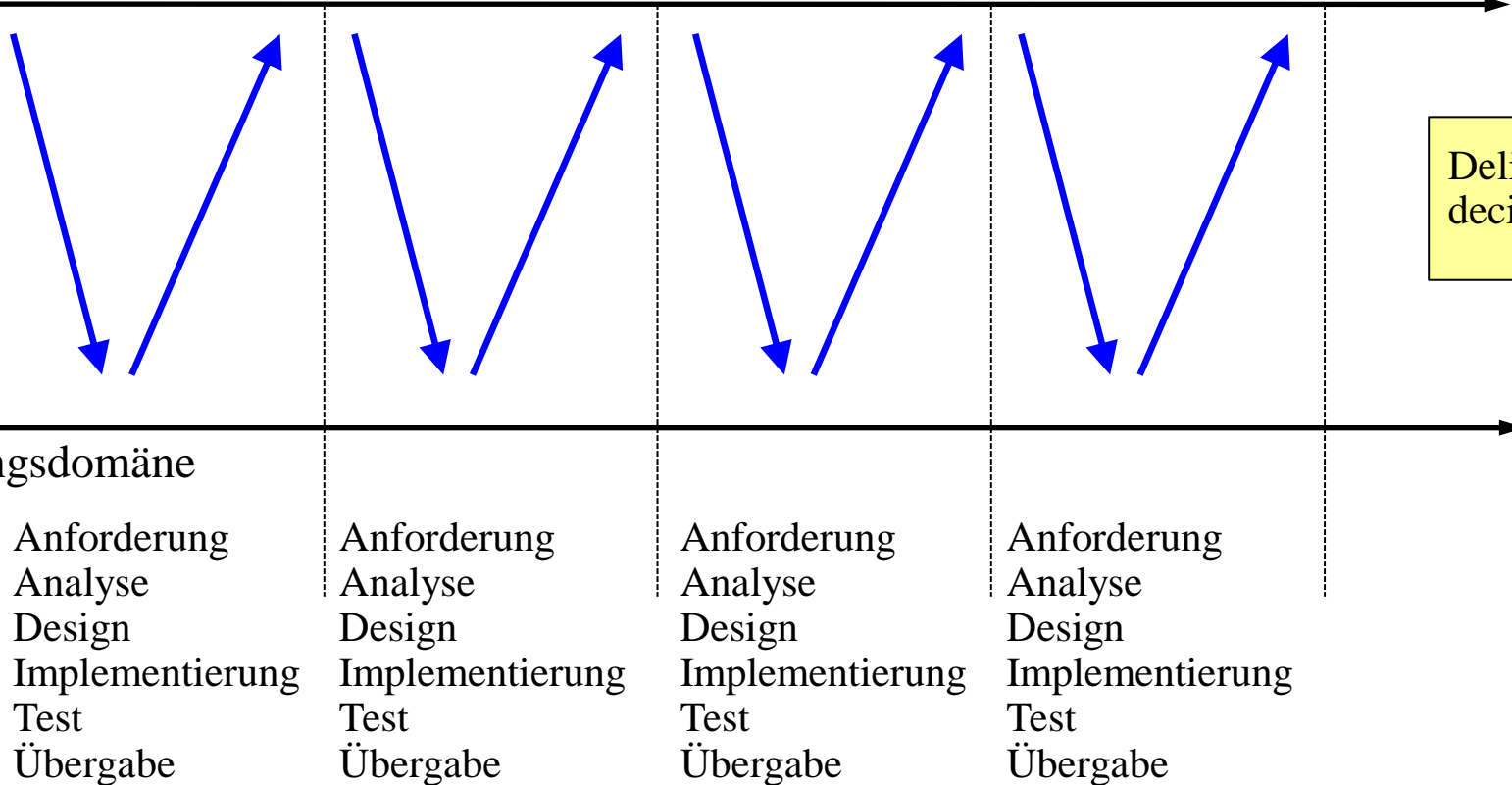
Das Problem oder das Verständnis ändert sich

Exkurs: Timeboxing

● Softwareentwicklung als empirischer Prozess.

Problemdomäne

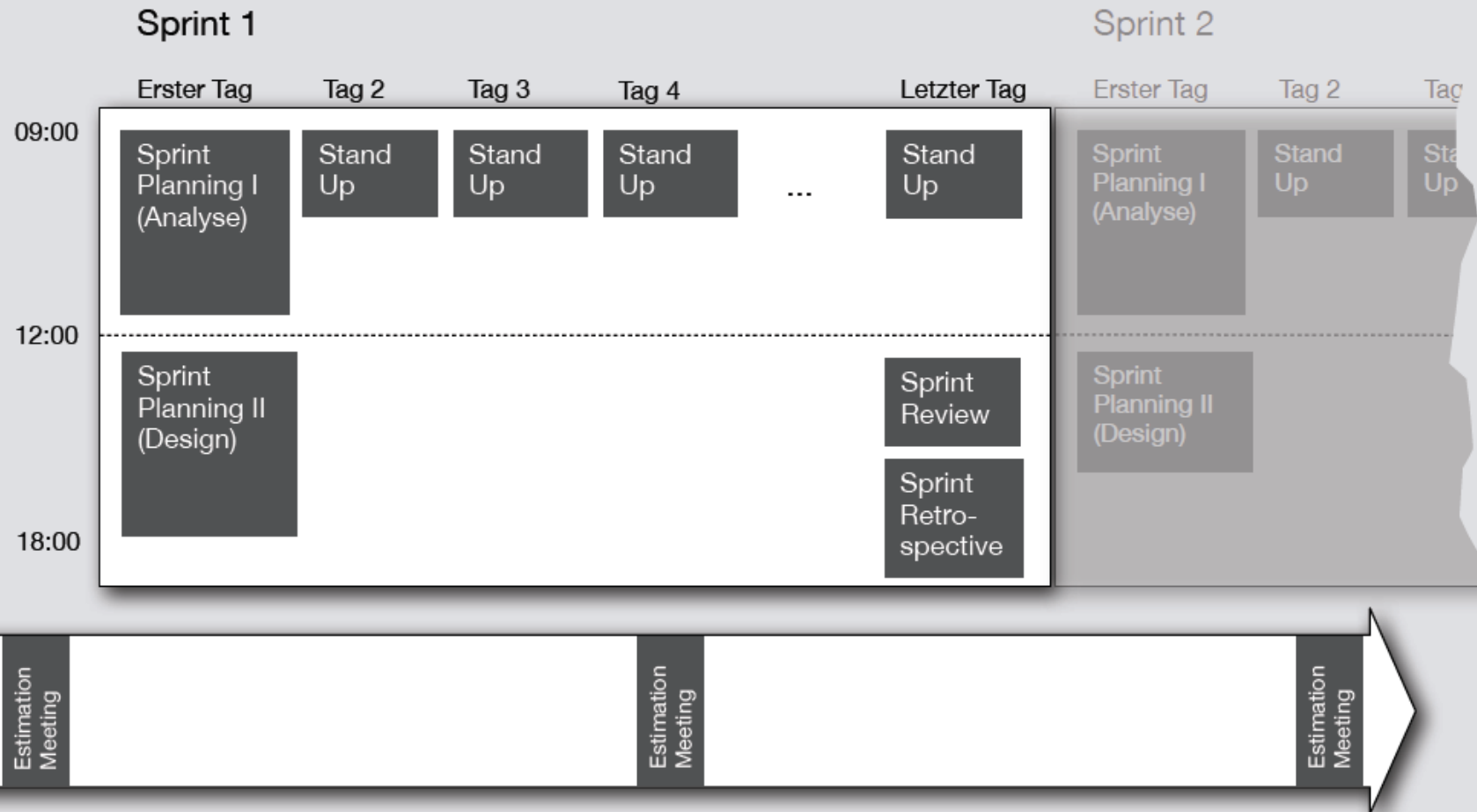
Lösungsdomäne



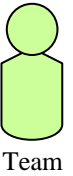
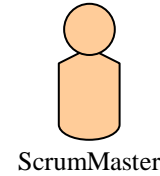
Deliver early,
decide late!

→ Interdomänen-Kommunikation

SCRUM Zeremonien im Sprint



Daily Scrum



● Ziel:

- Überblick über den Fortschritt der einzelnen Mitglieder schaffen
- Hindernisse diskutieren & beseitigen

● Inhalt: Jeder Teilnehmer beantwortet der Reihen nach drei Fragen.

- Was habe ich seit dem letzten Daily Scrum getan?
- Woran arbeite ich bis zum nächsten Daily Scrum?
- Was hat mich bei meiner Arbeit behindert?

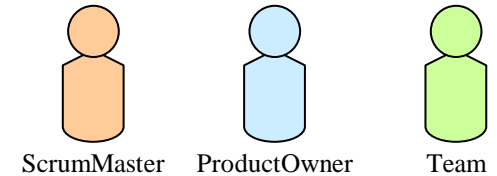
● Besonderheit:

- ScrumMaster notiert Hindernisse im Impediment Backlog.

● Timebox:

- Etwa 15 Minuten langes Standup-Meeting

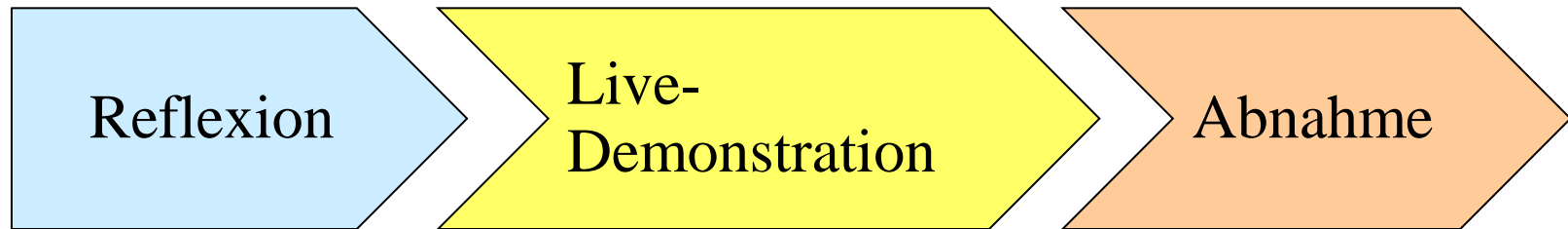
Sprint Review Meeting



● Ziel:

- Abnahme der Arbeitsfortschritte durch den ProductOwner

● Ablauf:



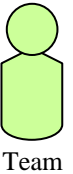
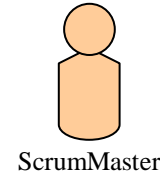
● Wichtig:

- Keine reine Marketingvorstellung – simple Demonstration der implementierten Anforderungen
- In der Abnahme wird vom ProductOwner der Sprint-Endbericht erstellt.

● Timebox:

- Je nach Größe des Sprints 2 bis 4 Stunden

Sprint Retrospective Meeting



● Idee

- Erfahrung durch Feedbackschleife

● Ziel:

- Zusammenarbeit des Teams verbessern
- Anwendung des Scrum-Prozesses optimieren
- Produktivität und Softwarequalität steigern

● Timebox:

- zwischen 2 und 3 Stunden

● Ablauf:

- Ein häufiges Verfahren ist "Mad Sad Glad": Jeder schreibt die drei wichtigsten positiven und die drei wichtigsten negativen Vorkommnisse auf Karteikarten.

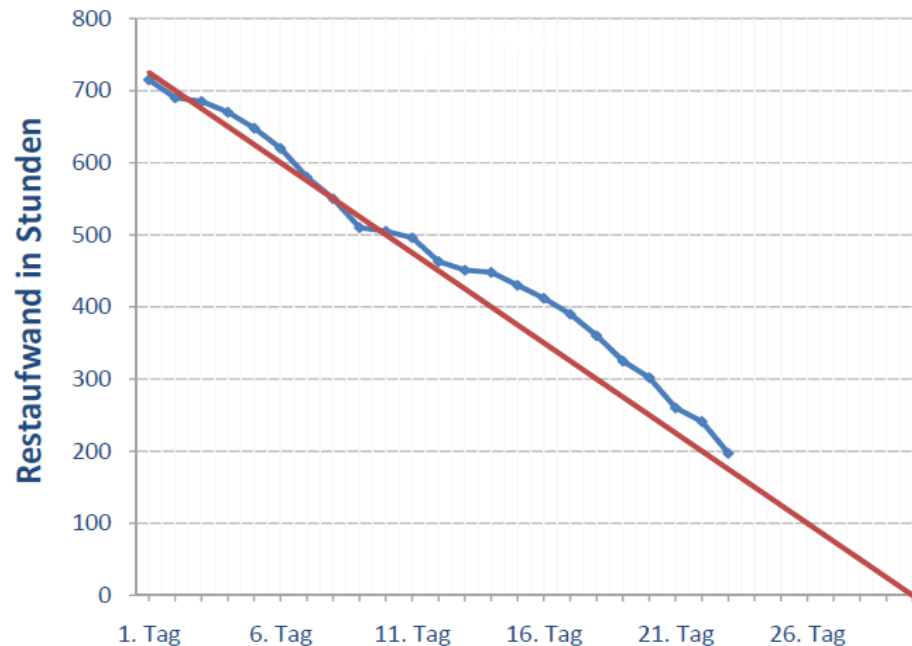
● Ergebnis:

- einige konkrete Verbesserungsmaßnahmen, welche optimalerweise die "SMART"-Kriterien erfüllen.

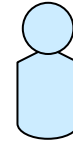
Artefakte in Scrum

● Der Sprint-Burndown-Chart:

- Ziel: visualisiert Arbeitsfortschritt im aktuellen Sprint



Artefakte in Scrum

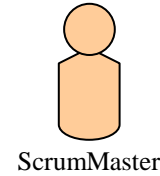


ProductOwner

● **Der Sprint-Endbericht:**

- Idee:
 - Am Ende eines jeden Sprints (vor der Sprint-Retrospektive) erstellt der ProductOwner beim Sprint-Review den Sprint-Endbericht.
- Aufbau:
 - Der Sprint-Endbericht fasst den Sprint-Verlauf zusammen. Enthalten ist die Abnahme umgesetzter Anforderungen, Verweise auf Testprotokolle und Qualitätsmetriken (z.B. Testabdeckung), der Sprint-Burndown-Bericht und Hindernisbericht.
 - Der Sprint-Endbericht sollte maximal zwei DIN-A4-Seiten lang sein.
- Wichtig:
 - Als abgenommen gelten nur 100 % fertig umgesetzte Anforderungen. "Fast fertig" oder "zu 99 % fertig" genügt nicht.
 - Dies beinhaltet auch Modultests, Integrationstests und funktionale Blackbox-Tests sowie Architektur-, Design- und Benutzerdokumentation.

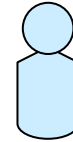
Artefakte in Scrum



● **Die Hindernisliste (Impediment Backlog):**

- Idee:
 - Das Team kümmert sich nur um die Umsetzung der Anforderungen. Für (die Behebung) von Hindernissen ist der ScrumMaster verantwortlich
- Aufbau:
 - In den Daily-Scrum-Besprechungen sammelt der ScrumMaster auftretende Hindernisse ("Impediment") in der Hindernisliste.
 - Der Product Owner berichtet im Sprint-Endbericht über die Hindernisse.
- Wichtig:
 - Der ScrumMaster versucht die Hindernisse aufzulösen.

Artefakte in Scrum



ProductOwner

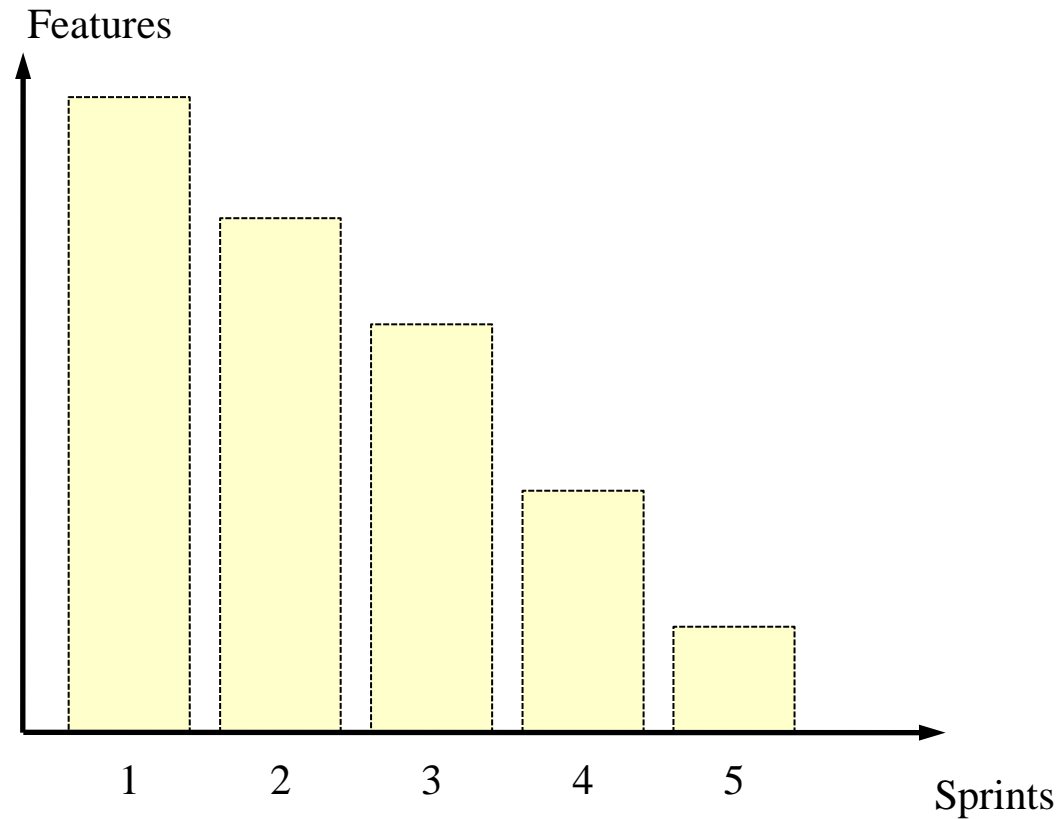
● **Der Releaseplan:**

- Idee:
 - Der Releaseplan bietet eine Übersicht zum Zeit- und Kostenrahmen.
 - Der Product Owner erstellt den Releaseplan und aktualisiert ihn in jedem Sprint.
- Inhalt:
 - Er enthält im Groben die Reihenfolge der Umsetzung der Anforderungen und die erwartete Anzahl Sprints.
 - Anfangs wird nur im Groben geplant.
 - Im Laufe des Projekts wird regelmäßig iterativ präzisiert.
- Wichtig:
 - Dabei wird insbesondere auf Risiken eingegangen!

Artefakte in Scrum

● Der Product-Burndown-Chart:

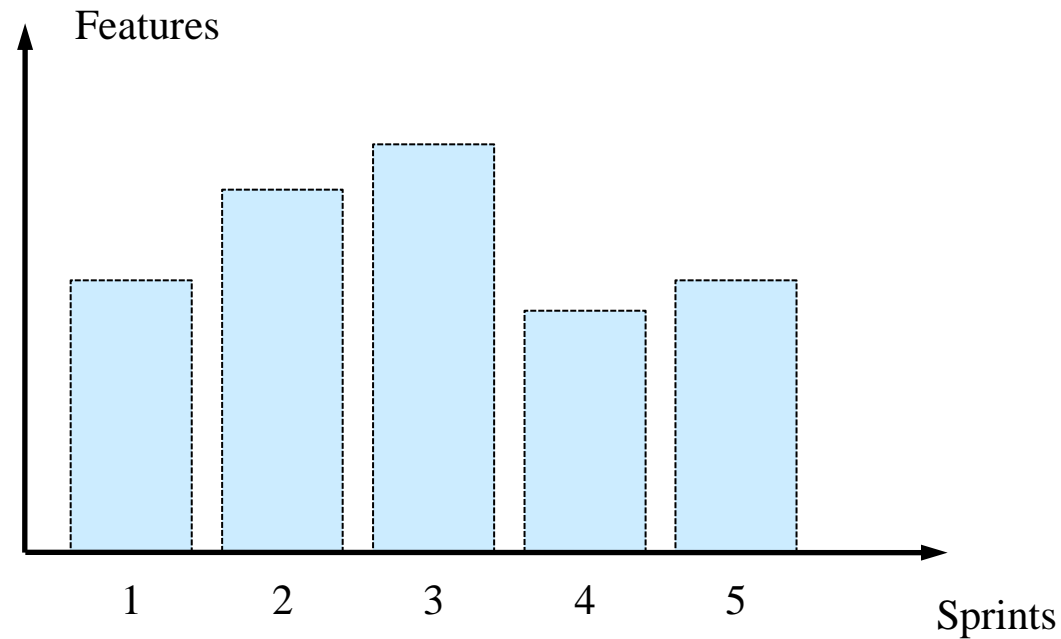
- Ziel: Visualisiert die noch auszulieferenden Anzahl von Features pro Sprint
- Nur fertiggestellte Features [Story Points]



Artefakte in Scrum

● Der Velocity-Chart:

- Ziel: Visualisiert die Menge der ausgelieferten Features pro Sprint
- Macht die Entwicklungsgeschwindigkeit des Teams klar
- Nur fertiggestellte Features [Story Points]



Artefakte in Scrum



ProductOwner

● **Der Releasebericht:**

- Idee:
 - Der Releasebericht ermöglicht die Verfolgung des Projektfortschritts und den Vergleich des tatsächlichen Fortschritts mit dem geplanten.
- Aufbau:
 - Er kann beispielsweise in Form eines Release-Burndown-Berichts, als Entwicklungsgeschwindigkeitsbericht (Velocity-Chart) und/oder als Themenpark (Fertigstellungsgrad gruppiert nach Themen) ausgeführt werden.

SCRUM: Zusammenfassung

● Vorteile:

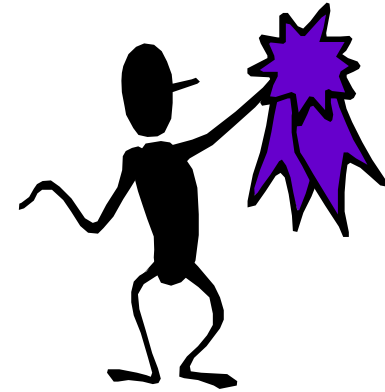
- Einfache Regeln
- Wenige Rollen
- Selbstorganisierte, interdisziplinäre Teams
- Iteratives Vorgehen

● Nachteile:

- Risiko dominanter Teammitglieder
- Zeitverlust durch häufige Meetings
- Product Backlog abhängig von eigenen Interessen der einzelnen Teammitglieder

● Agiles Manifest:

- Menschen wichtiger als Prozesse: ja
- Laufende Software wichtiger als Doku: ja
- Zusammenarbeit mit dem Kunden wichtiger als Vertragsverhandlungen: ja
- Veränderungen begrüßen statt Planverfolgung: ja!



Vielen Dank!