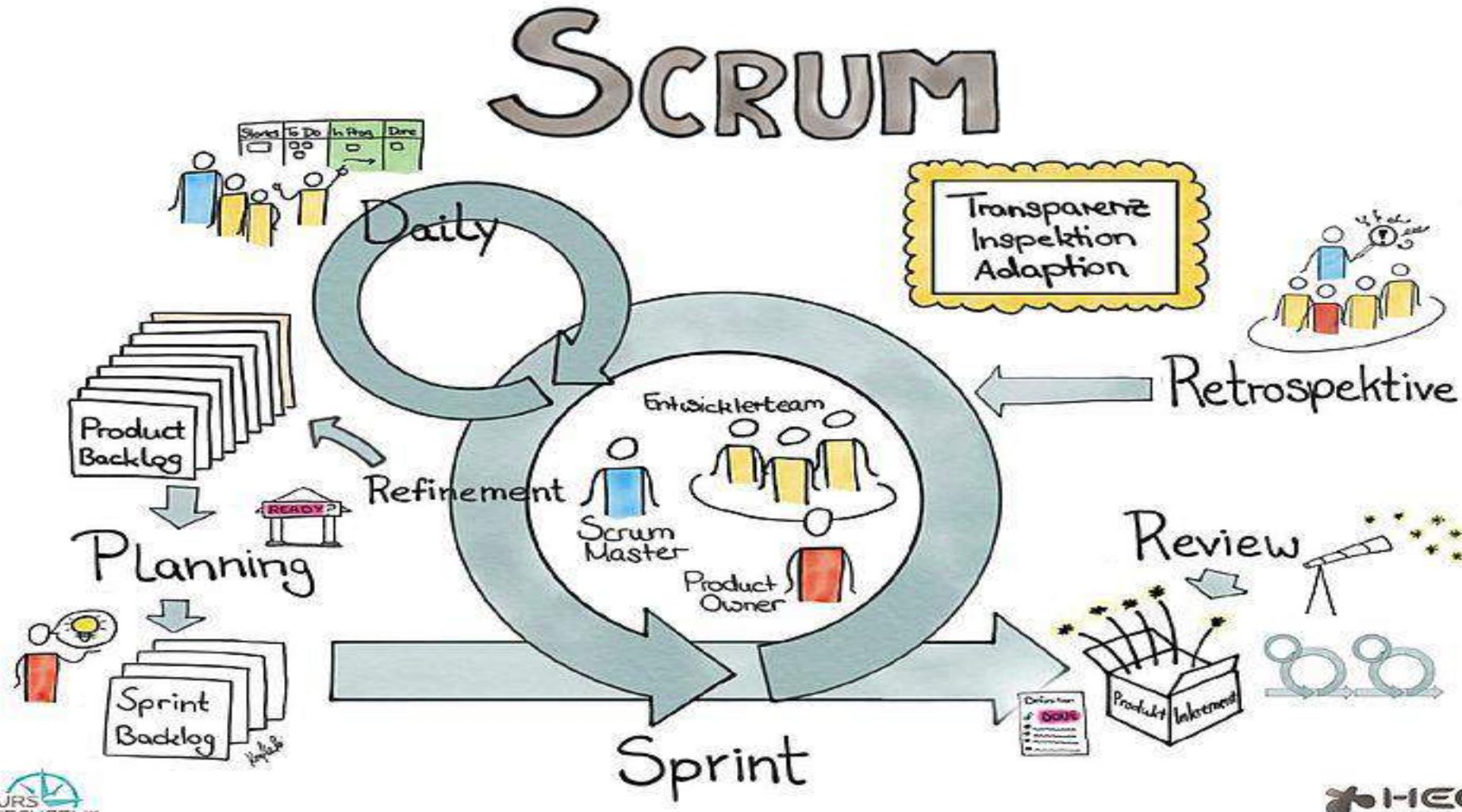


PROJEKTMANAGEMENT MIT SCRUM



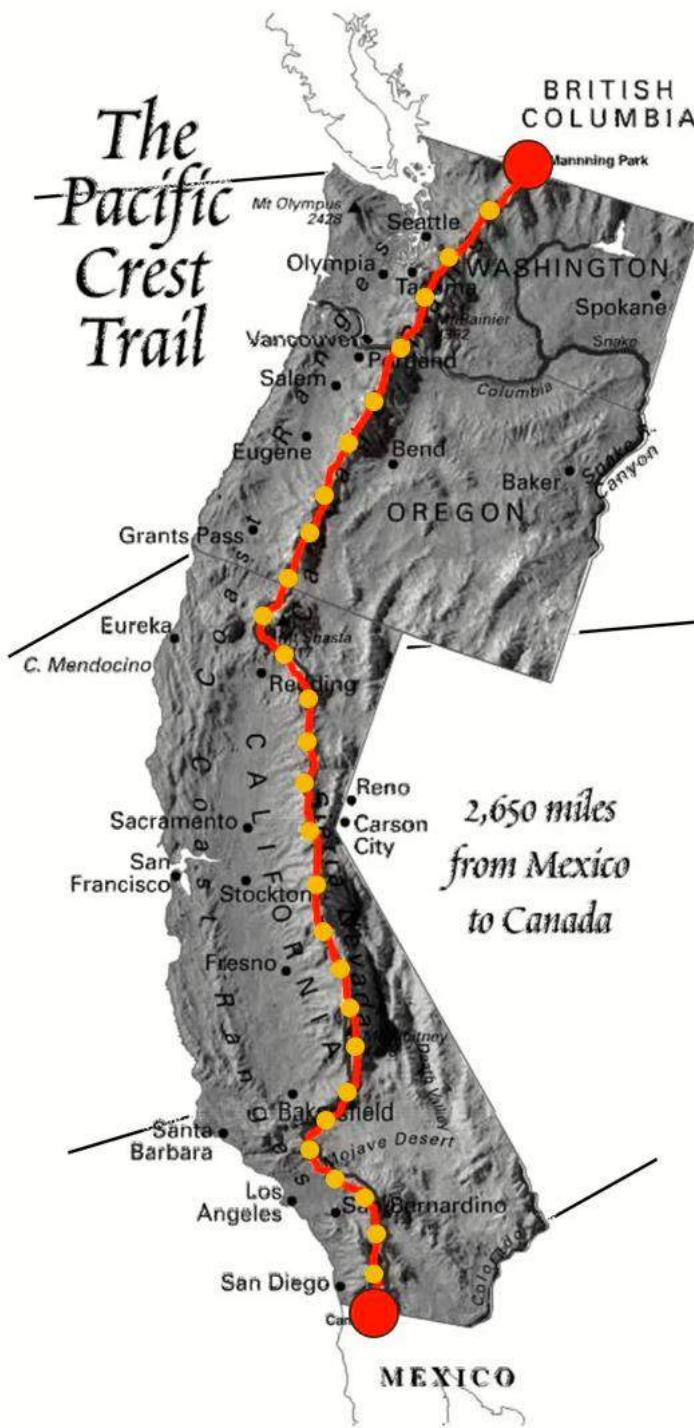
VERGLEICH AGIL – KLASISCH AN EINEM BEISPIEL

WANDERUNG VON MEXIKO NACH KANADA



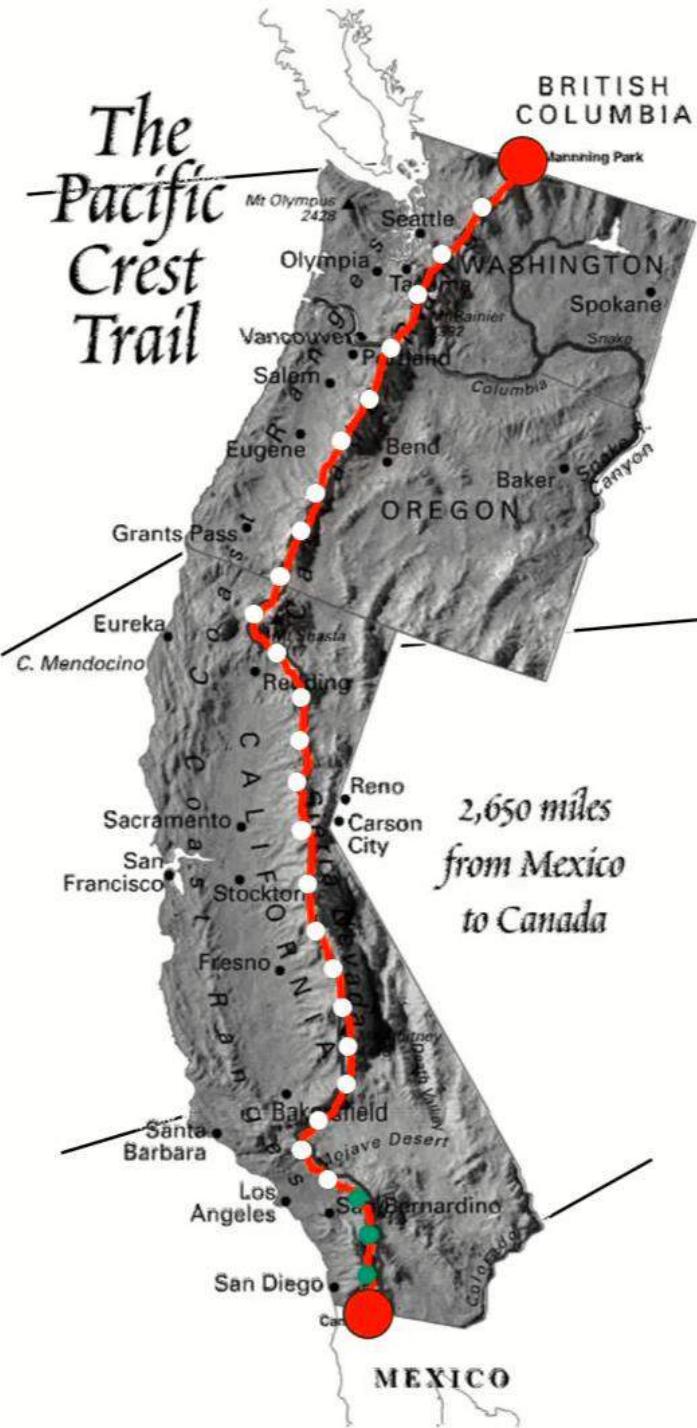
VERGLEICH AGIL – KLASSISCH AN EINEM BEISPIEL

KLASSISCH:
ETAPPEN MIT
MEILENSTEINEN



VERGLEICH AGIL – KLASISCH AN EINEM BEISPIEL

AGIL:
NUR DIE ERSTEN
ETAPPEN WERDEN
GEPLANT



VERGLEICH AGIL – KLASISCH AN EINEM BEISPIEL

AGIL:
NUR DIE ERSTEN
ETAPPEN WERDEN
GEPLANT



VERGLEICH AGIL – KLASISCH AN EINEM BEISPIEL

AGIL:
IN DAS ZWEITE
ETAPPENPAKTET
FLIEßen DIE
ERFAHRUNGEN DES
ERSTEN EIN

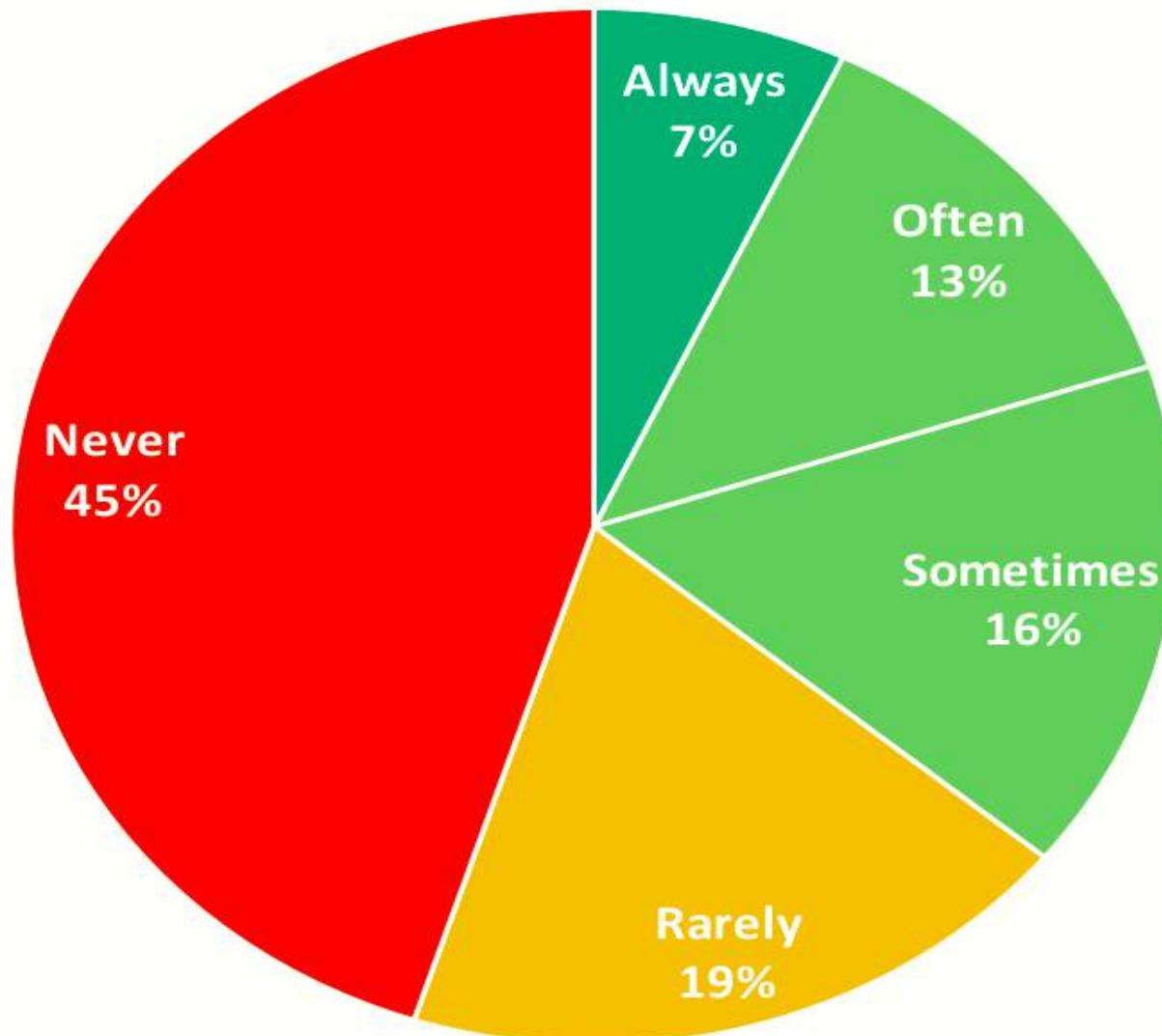


DURCHFÜHRUNG NORDAMERIKA EXPEDITION MIT SCRUM

ZUSAMMENFASSUNG

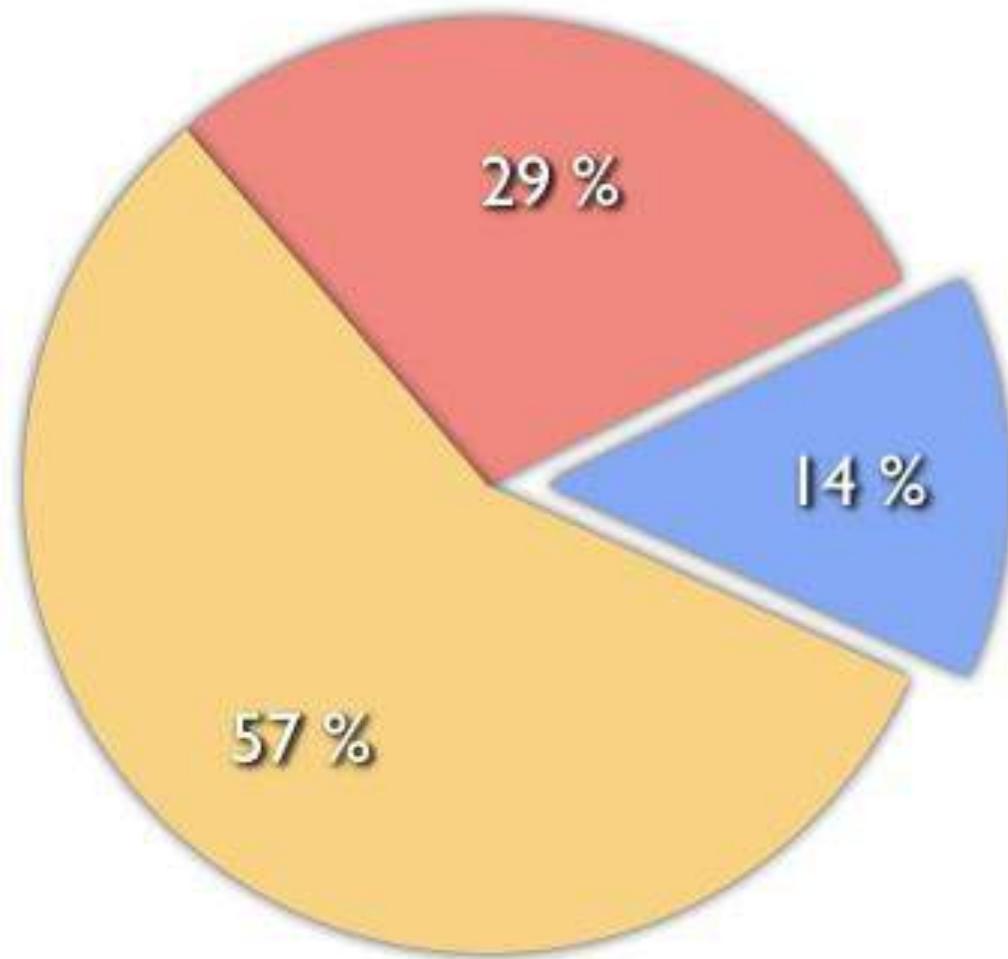


WELCHE FEATURES EINER FERTIGEN SOFTWARE WERDEN TATSÄCHLICH GENUTZT?

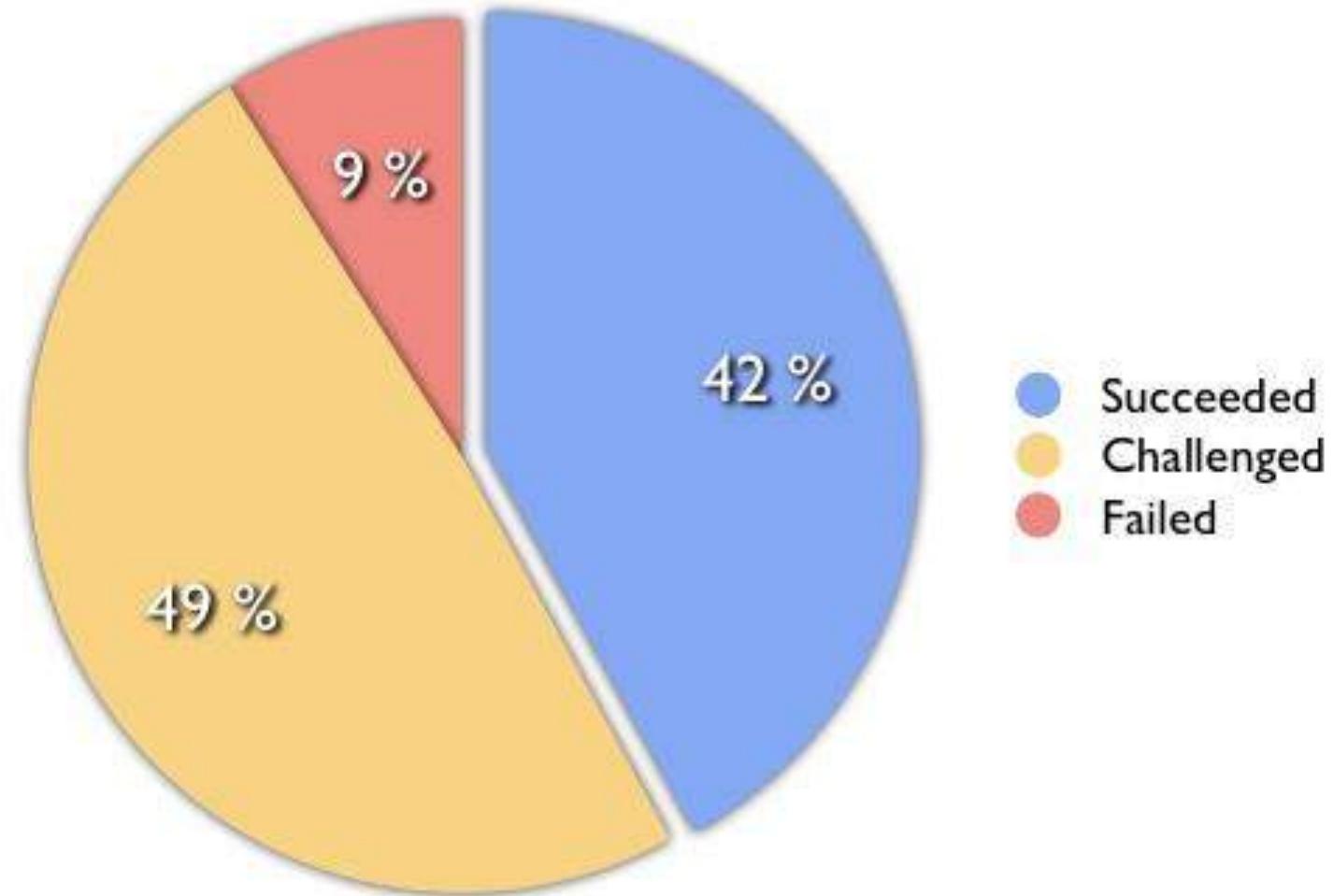


ERGEBNIS PROJEKTE WASSERFALL / AGIL

Waterfall



Scrum



VERGLEICH KLASSISCH / AGIL

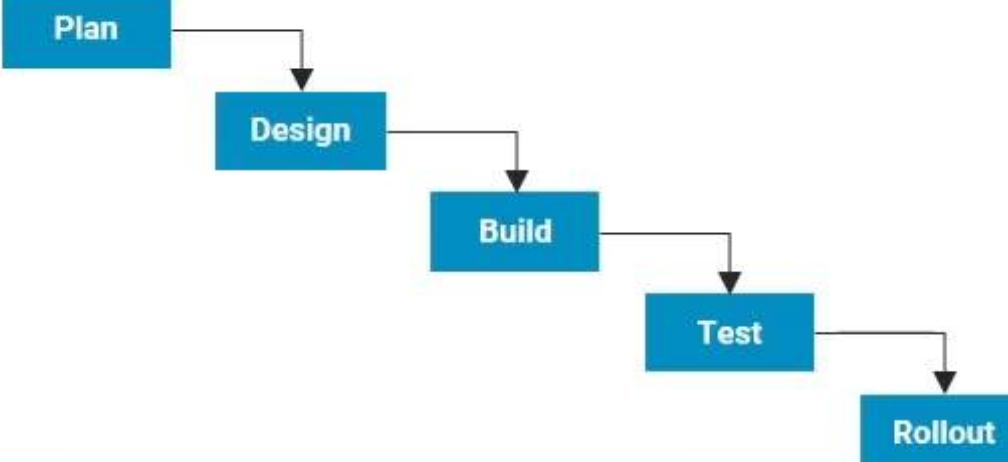
Das Vorgehensmodell im Vergleich

Klassische und agile Methoden unterscheiden sich insbesondere in der Herangehensweise

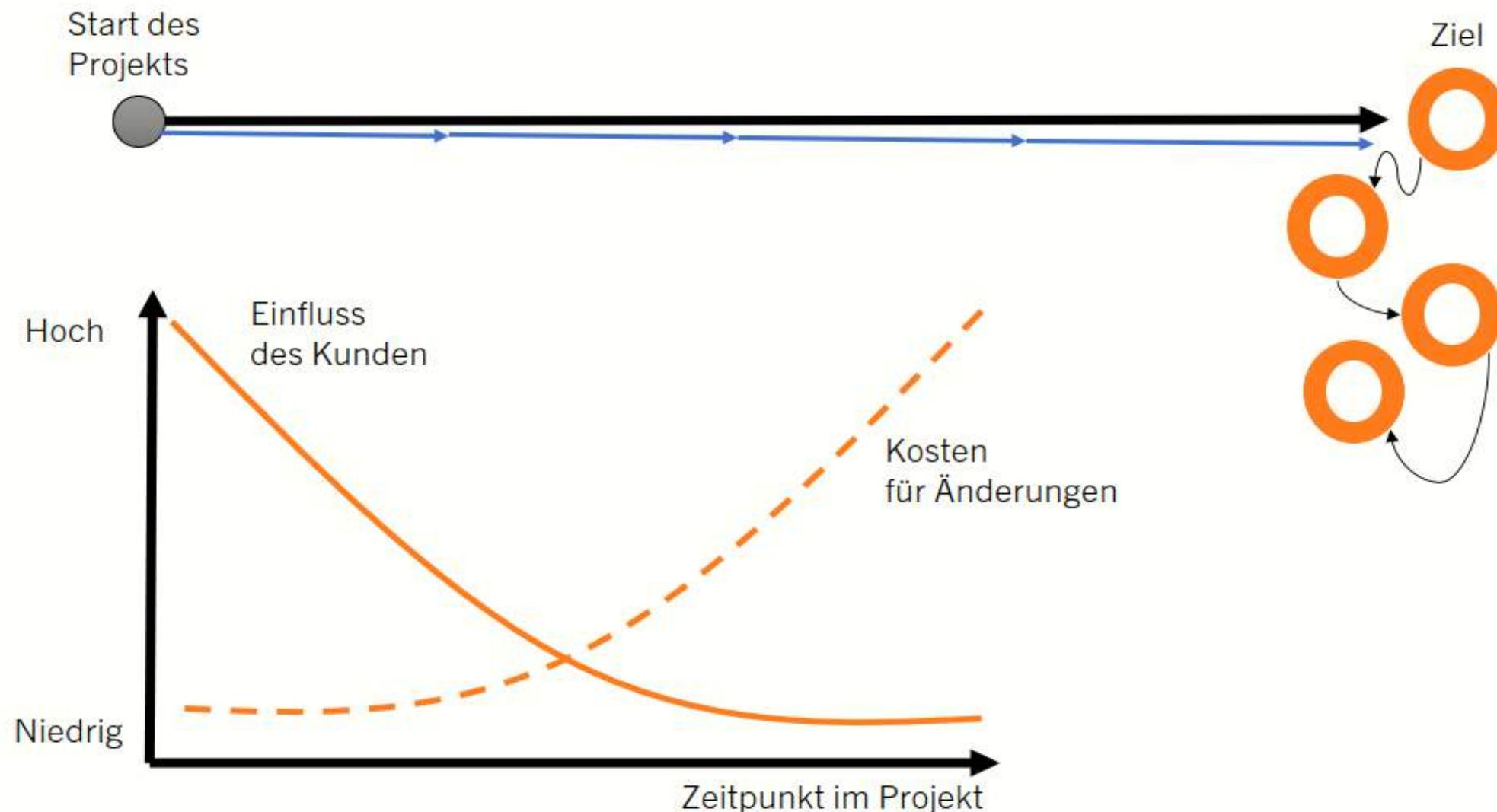
Wasserfall (klassisch)

- **Intensive Auseinandersetzung** mit allen Anforderungen und eine **detaillierten Planung aller Anforderungen** zu Projektbeginn
- Nachdem **alle Anforderungen geplant und konzipiert** wurden erfolgen **Umsetzung, Test und Lieferung**
- **Änderungen** im Projekt wirken meist primär auf den Einsatz von **Kosten** und, oder **Zeit**
- **Einzelne Phasen** dauern oft mehrere **Monate**

Wasserfall (klassisch)



PROBLEME BEIM KLASSISCHEN ARBEITEN



ÄNDERUNGSKOSTEN FORTSCHREITENDER PROJEKTE IM KLASSISCHEN PROJEKTMANAGEMENT

Beispiel:



Zu Beginn eines Brückenbau-Projektes wird man versuchen, die Vorstellungen der verschiedenen Stakeholder (Ämter, Bürger, Experten, Verbände usw.) zu berücksichtigen. Später ist das schon schwieriger: Sind dann die ersten Brückenpfeiler gesetzt, wird eine Veränderung z.B. der Nutzlast oder des Verlaufs der Brücke sehr teuer. Die Möglichkeit der Stakeholder, zu diesem Zeitpunkt noch Einfluss zu nehmen, wird damit deutlich begrenzt.

VERGLEICH KLASSISCH / AGIL

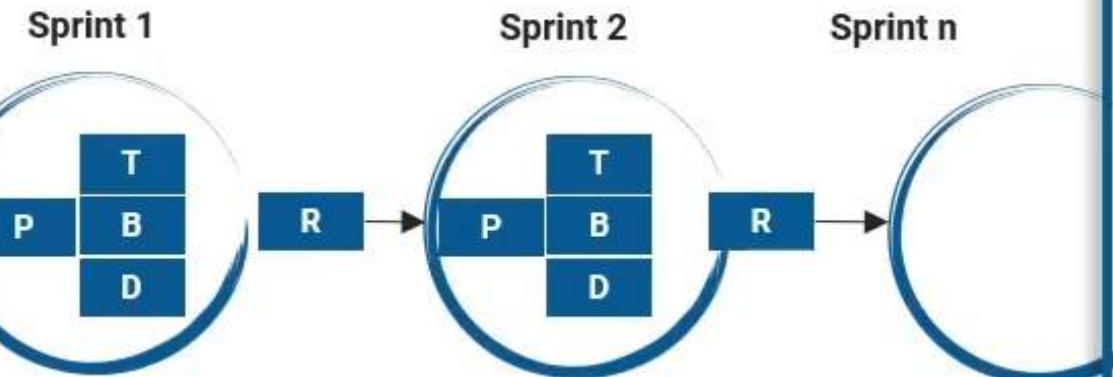
Das Vorgehensmodell im Vergleich

Klassische und agile Methoden unterscheiden sich insbesondere in der Herangehensweise

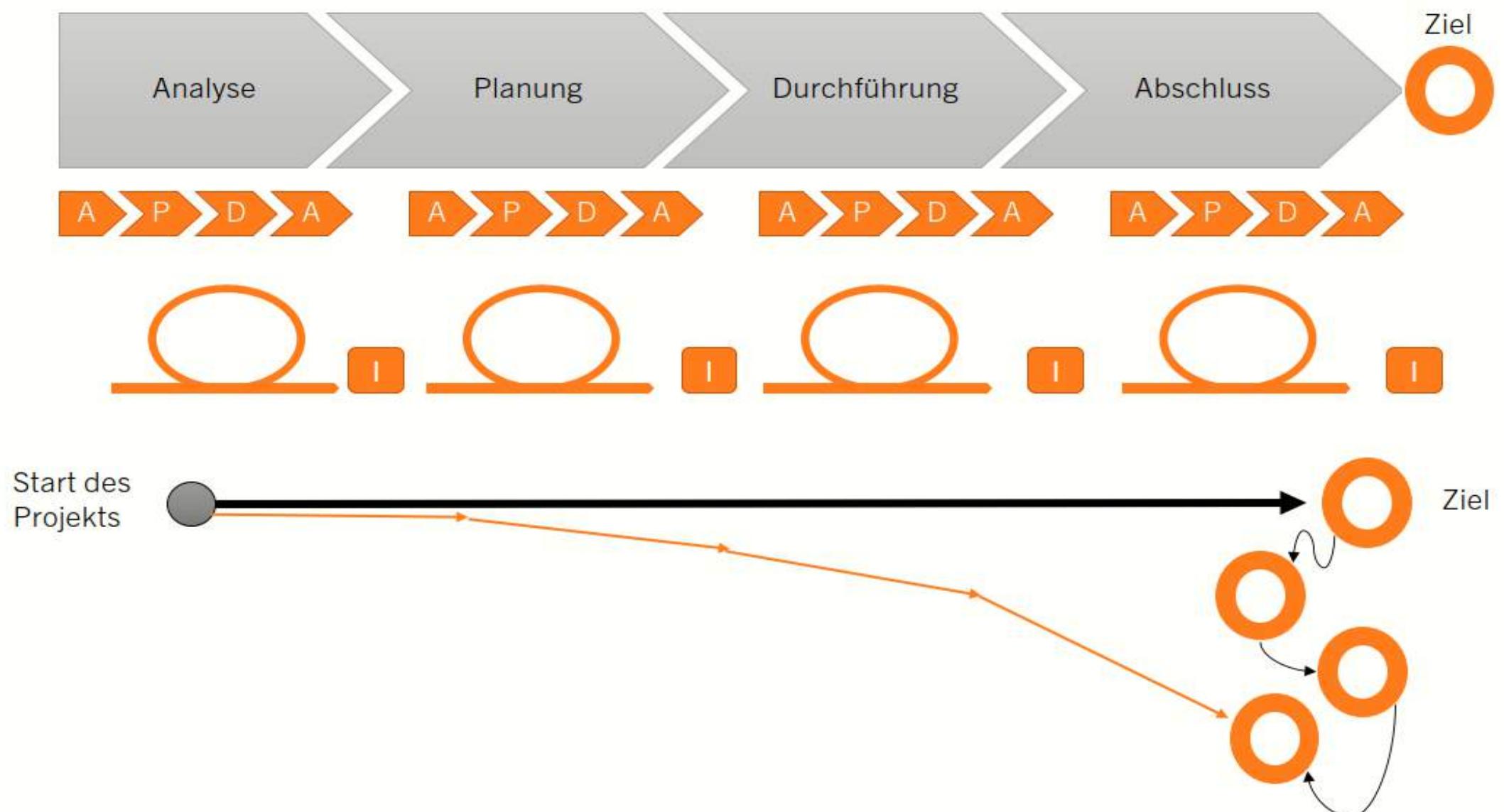
Scrum (agil)

- Fokus auf **frühe erste Ergebnisse und deren Review**
- **Anforderungen werden kontinuierlich aufgenommen**, grob beschrieben und frühzeitig priorisiert.
- **Anforderungen mit hoher Priorität** werden dann zeitnah **detailliert** und möglichst innerhalb eines „Sprints“ (i.d.R. 2-4 Wochen) geplant und **umgesetzt**
- Im nächsten Sprint werden dann erneut die jeweils höchsten Anforderungen ausgeplant und umgesetzt
- Ein **Sprint umfasst Planung, Design, Entwicklung und Test und Bereitstellung** der jeweiligen Funktionalität (Anforderung)

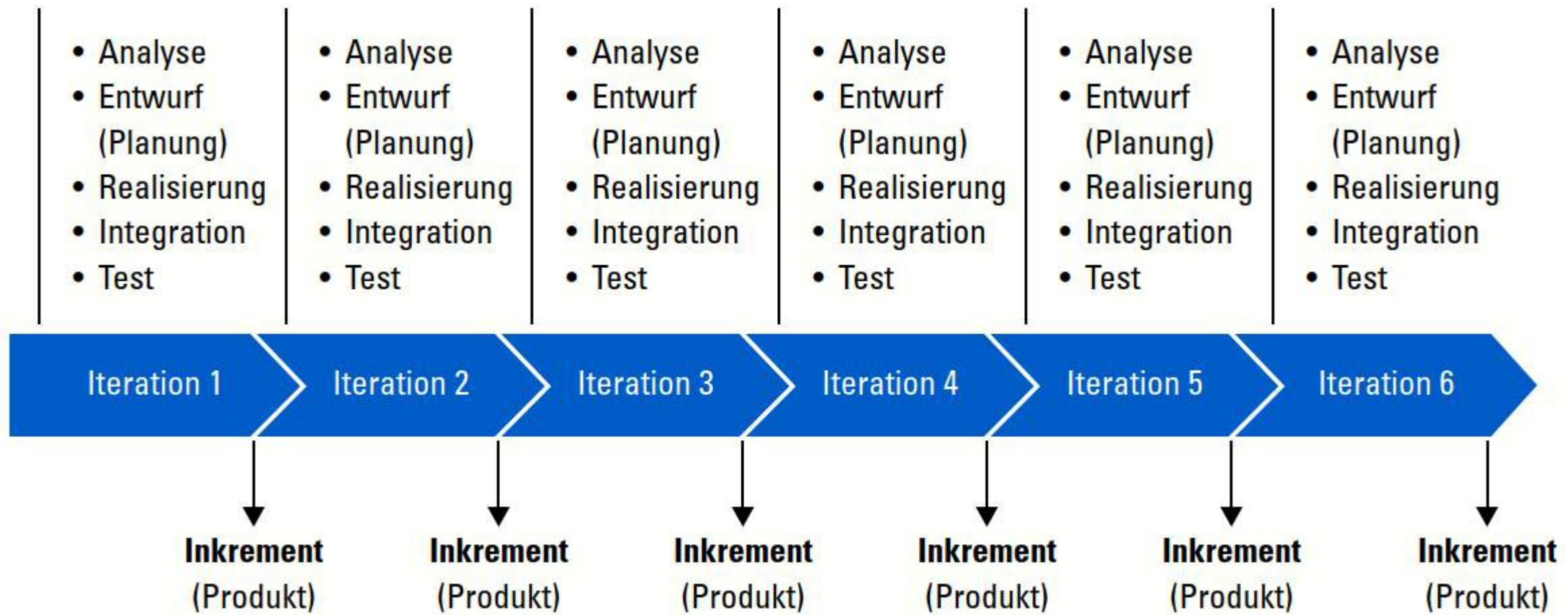
Scrum (agil)



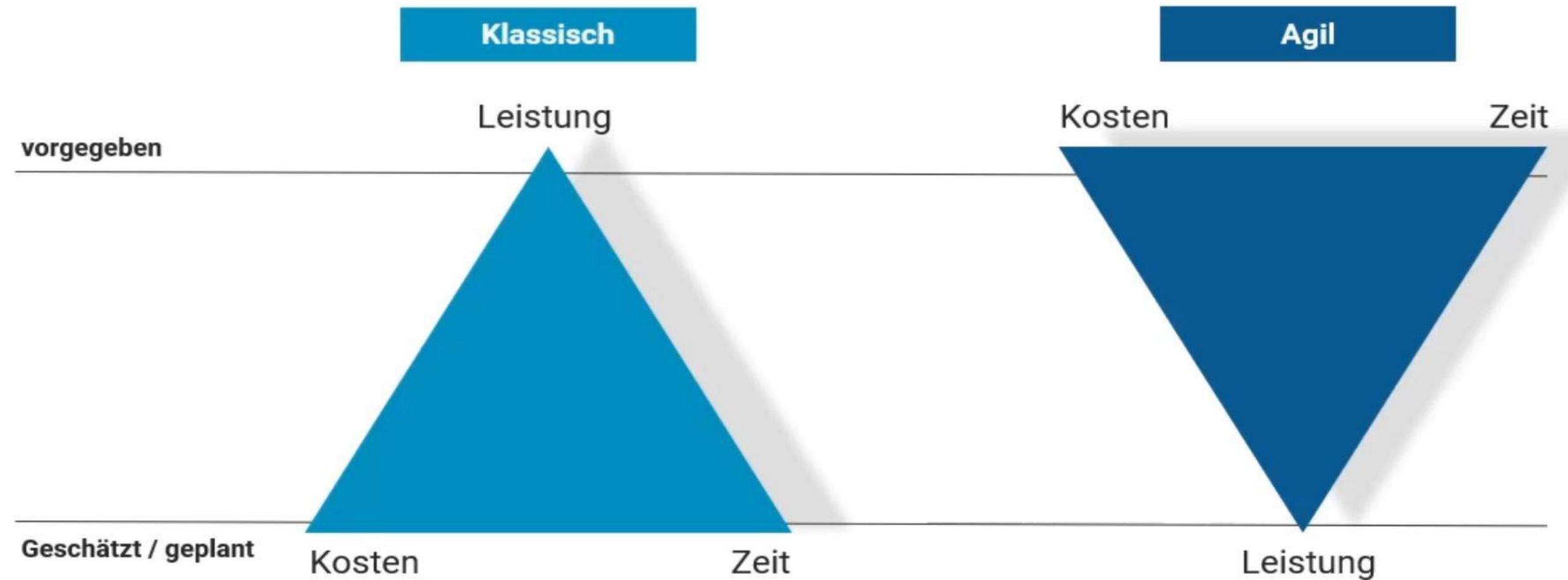
AGILES ARBEITEN



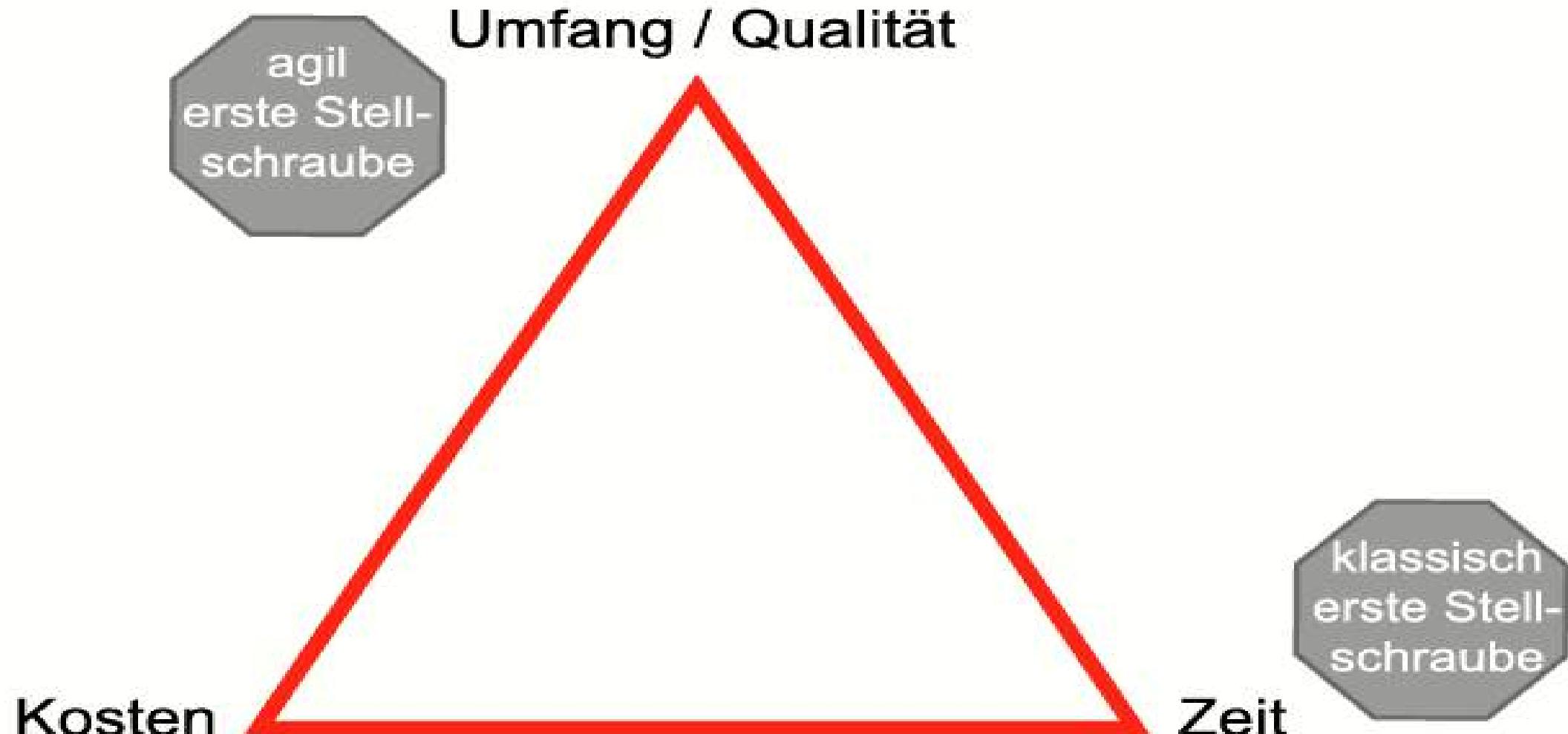
WARUM SCRUM?



DAS MAGISCHE DREIECK IM KLASSISCHEN UND AGILEN PROJEKTMANAGEMENT



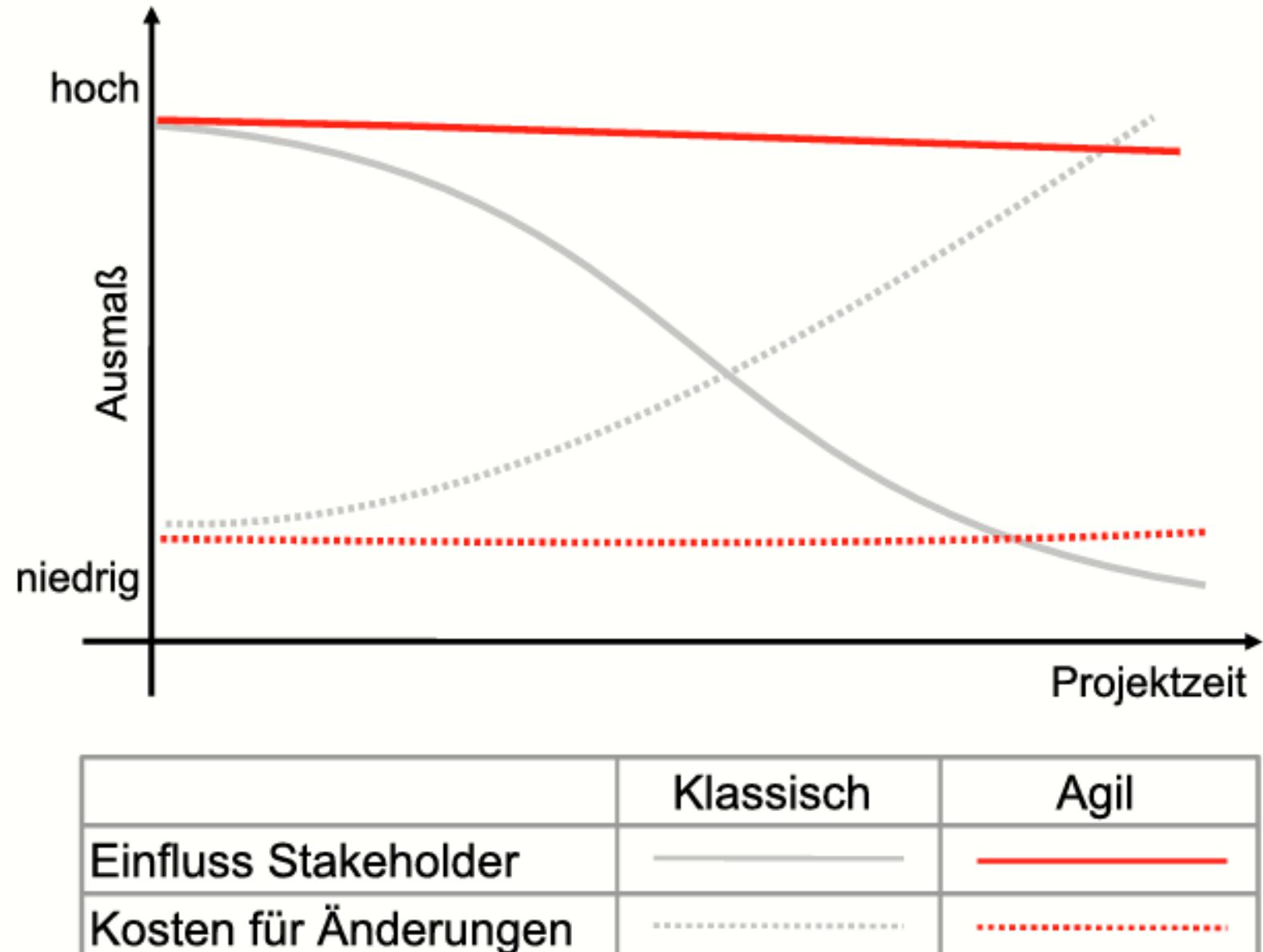
DAS MAGISCHE DREIECK AGIL VERSUS KLASSISCH



Die Stellschrauben im Zieldreieck: klassisch vs. agil

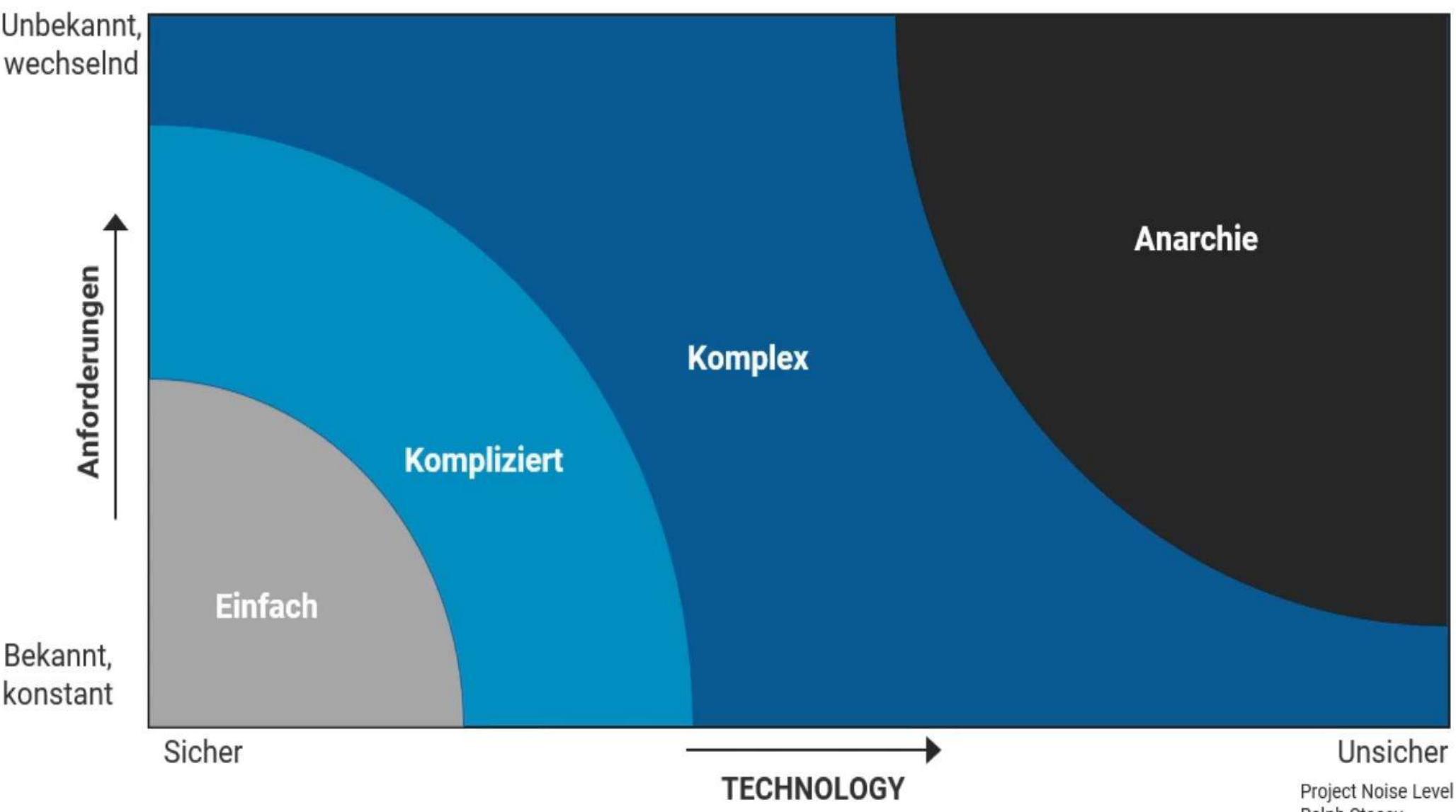
VERGLEICH ÄNDERUNGSKOSTEN UND STAKEHOLDER- EINFLUSS

KLASSISCH VERSUS AGIL



Veränderung im Projekt: klassisch versus agil

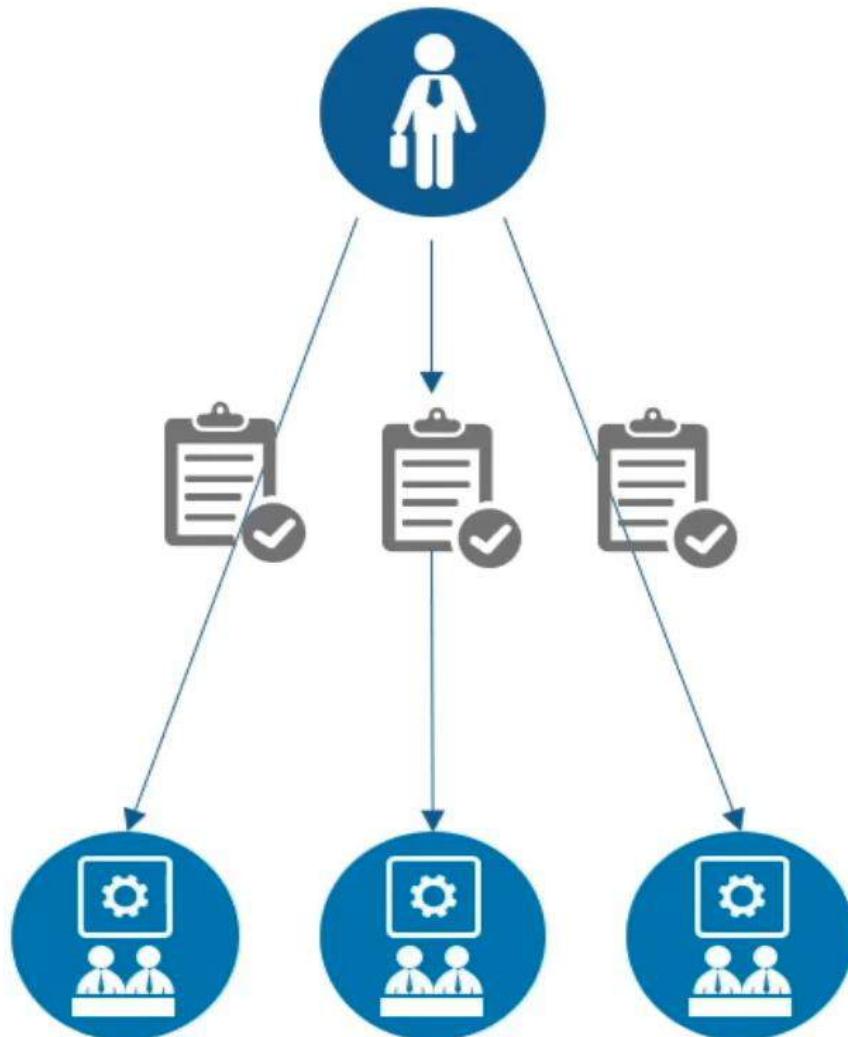
KLASSISCH
ODER
AGIL?



Unsicher

Project Noise Level
Ralph Stacey

HOL VS. BRINGSCHULD KLASSISCH: TOP-DOWN



Projektleiter

Verteilt Aufgaben an die Projektmitglieder, oder Teilprojektleiter. Steht hierarchisch über den anderen Mitgliedern des Projekts

Aufgaben / Pläne

In klassischen Projekten sind diese häufig bereits detailliert ausgearbeitet und müssen entsprechend der vorliegenden Pläne umgesetzt werden

Projektteam

Nimmt Aufgaben vom Projektleiter, oder Teilprojektleiter entgegen und setzt diese gemäß bestehender Pläne und Anweisungen um

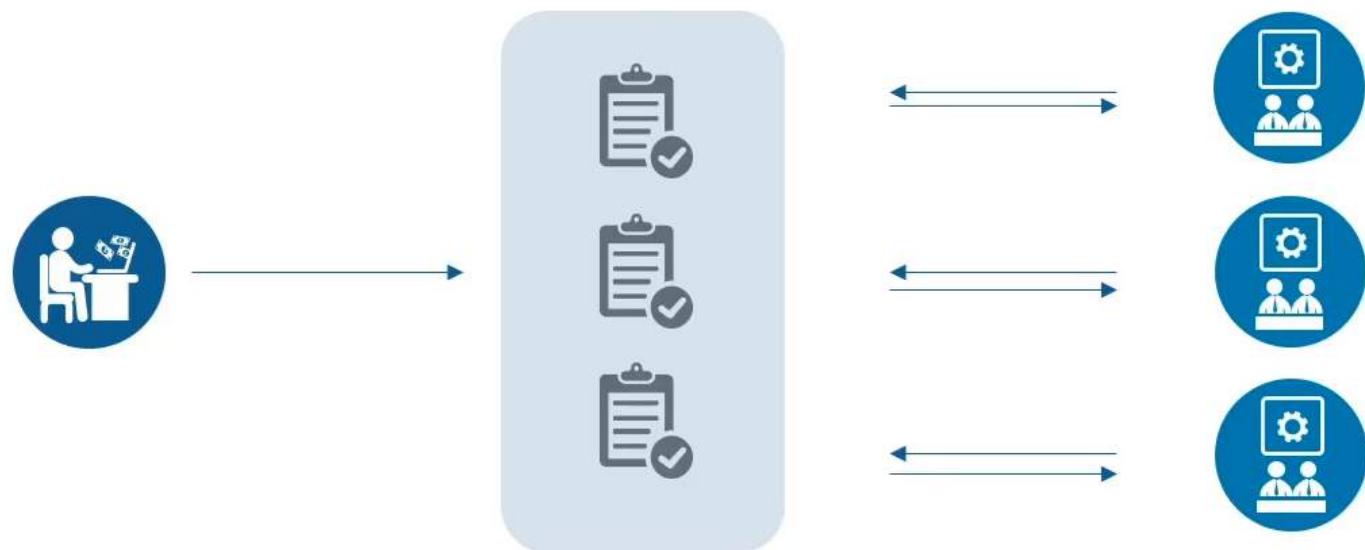
HOL VS. BRINGSCHULD

**AGIL: ENTWICKLUNGSTEAM
ZIEHT SICH DIE AUFGABEN**

Hol- versus Bringschuld

Scrum kennt keine Hierarchie und setzt auf die Einigung zwischen Product Owner und Entwicklungsteam, bei der Festlegung von Sprintinhalten

Scrum Aufgabenverteilung



Product Owner

Stellt seine Vision für den jeweiligen Sprint vor und nennt im Sprint Planning jene Anforderungen die dafür umgesetzt werden **sollten**

Sprint Planning & Anforderungen

Der Product Owner und das Entwicklungsteam besprechen im Sprint Planning die geplanten Anforderungen. Die Anforderungen sind noch nicht in konkrete Aufgaben heruntergebrochen. Ein Sprintziel wird vom Scrum Team vereinbart

Entwicklungsteam

Das **Entwicklungsteam entscheidet**, welche Anforderungen im kommenden Sprint **umsetzbar sind** und **committet sich** auf die Umsetzung eben jener Anforderungen. Das Entwicklungsteam ist eigenverantwortlich für die Erreichung des **Sprintziels** und bestimmt die hierzu operativ erforderlichen Tätigkeiten

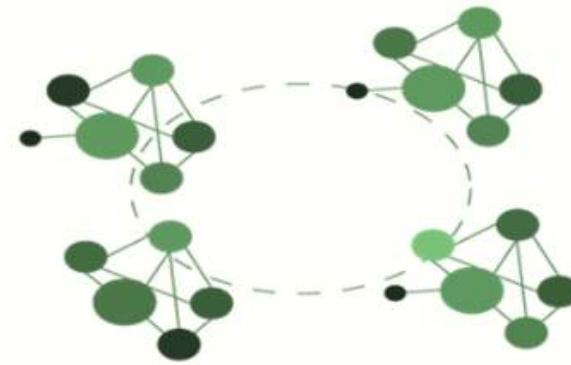
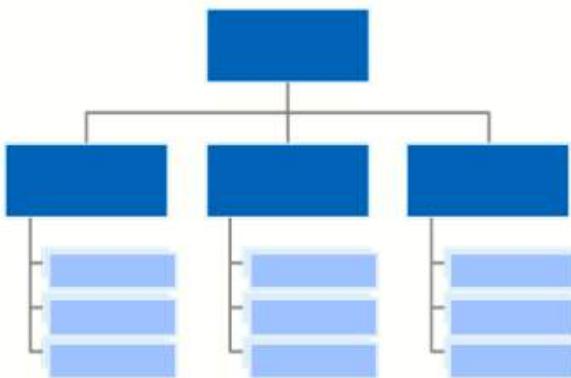
DIE ORGANISATION

Organisationsdesign

Klassisch (linear)

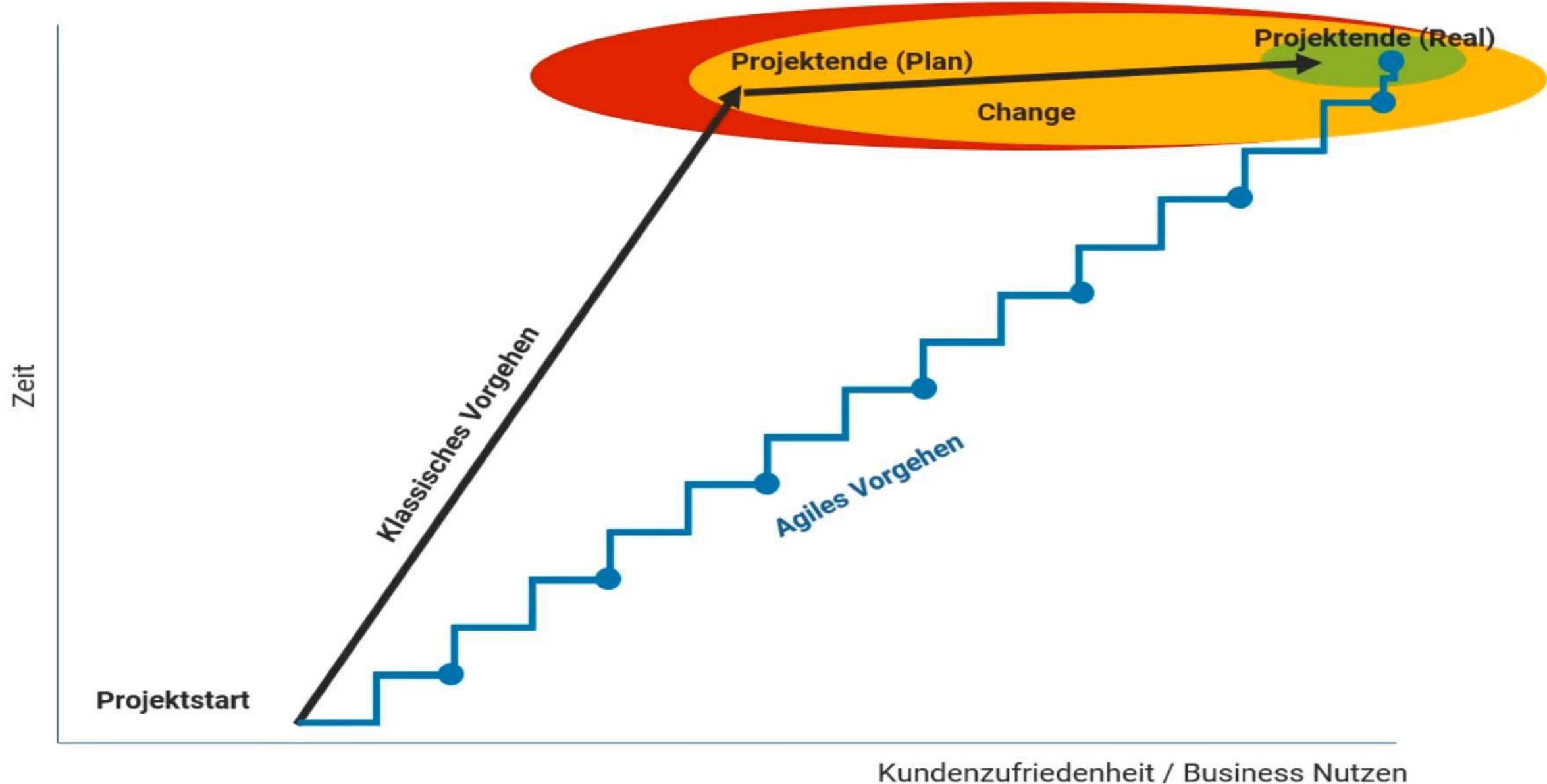
vs.

Agil



Struktur	<ul style="list-style-type: none">• Linear / fixiert	<ul style="list-style-type: none">• Team-Netzwerk / dynamisch
Ausgestaltung	<ul style="list-style-type: none">• Hierarchisch	<ul style="list-style-type: none">• Selbstorganisierend
Leadership	<ul style="list-style-type: none">• Top-down gem. Position	<ul style="list-style-type: none">• Individuell gem. fachlicher Rolle
Prozess	<ul style="list-style-type: none">• „Strategy and execution“	<ul style="list-style-type: none">• „Change and adapt“
Entscheidung	<ul style="list-style-type: none">• Top-down	<ul style="list-style-type: none">• Konsent-basiert (nicht Konsens)
Wertefokus	<ul style="list-style-type: none">• Prozesse / Verhandlung / Befolgung	<ul style="list-style-type: none">• Individuen / Kooperation / Interaktion

SIND AGILE PROJEKTE SCHNELLER?



ZUSAMMENFASSUNG

VERGLEICH KLASSISCH / AGIL

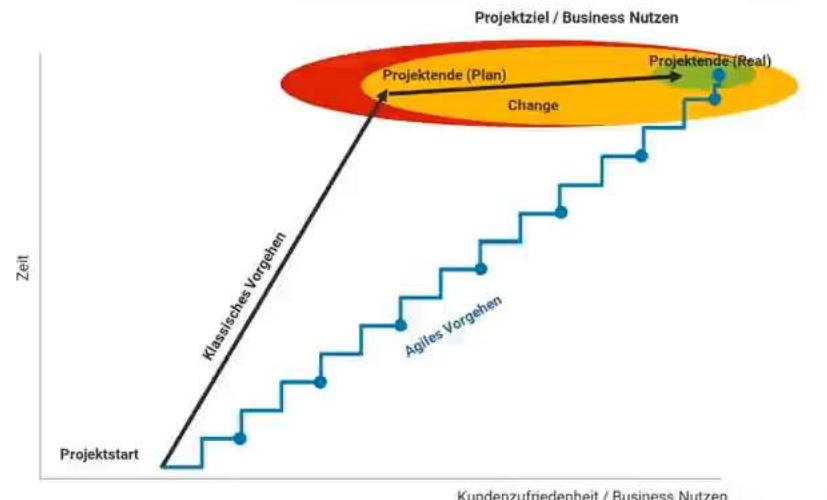
Philosophie

Klassisches und agiles Projektmanagement unterscheiden sich fundamental! Agiles Projektmanagement macht **keine festen Versprechungen** zu detaillierten Leistungszielen!



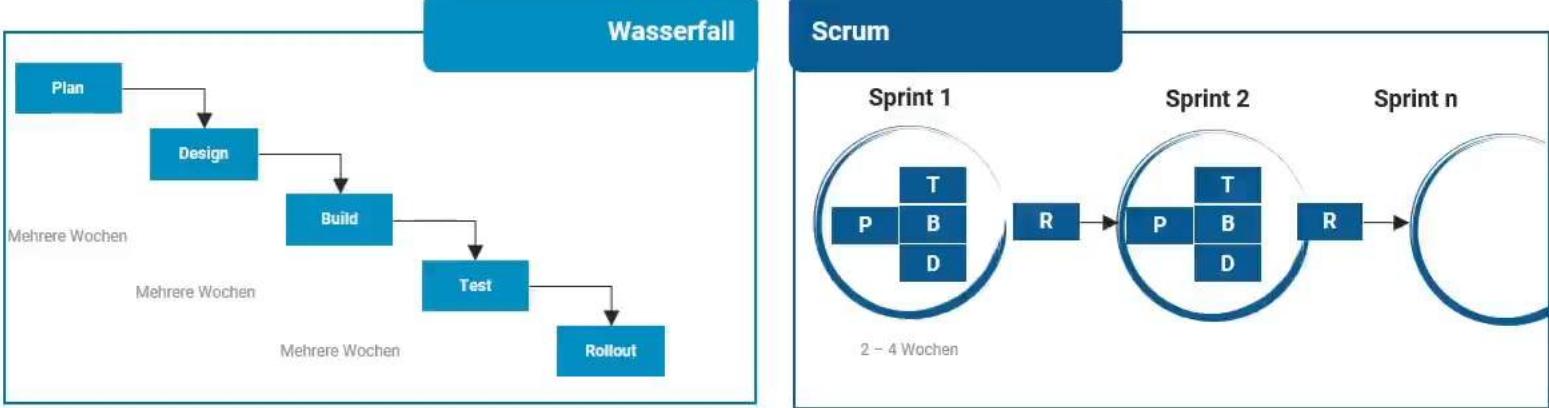
Sind agile Projekte insgesamt schneller?

Dies ist eine Frage der **Perspektive**:
 Agile Projekte liefern **schneller** einen ersten **messbaren Business Nutzen**. Eine vergleichbare Anzahl von Anforderungen, kann durch agile Vorgehensweisen aber **nicht grundsätzlich schneller** abgearbeitet werden.



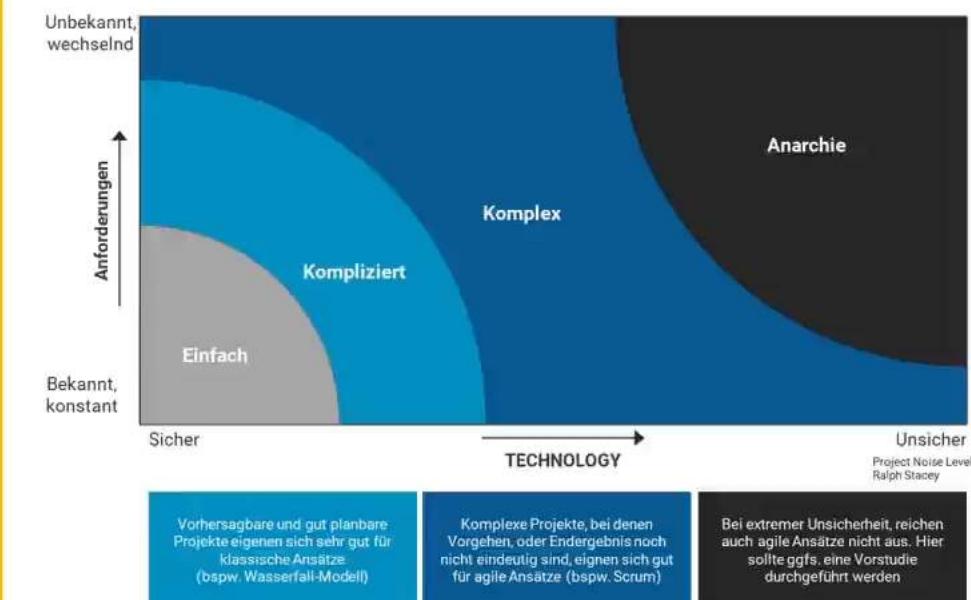
Ablauf

Scrum setzt auf viele kurze Iterationen (Sprints), mit einem kleinen Anteil von Anforderungen



Einsatz agiler Methoden

Agile Projekte scheitern, wenn keine Vision vorhanden ist!



Aufgabenverteilung

Agile Projekte kennen **keine Hierarchie** und profitieren von **intrinsischer Motivation** und der Umsetzung **agiler Werte**



DAS AGILE MANIFEST: VIER GRUNDWERTE

Individuen und Interaktionen **mehr**

als Prozesse und Werkzeuge

Software wird von Menschen gemacht. Prozesse und Werkzeuge sind wichtig, sollten aber primär dazu dienen, die Interaktion zwischen Individuen zu unterstützen als diese zu ersetzen.



Funktionierende Software

(Ergebnisse) mehr als umfassende

Dokumentation

Funktionierende und einsetzbare Projektergebnisse sind wichtiger als umfassende Dokumente, oder theoretische Konzepte.

Zusammenarbeit mit dem Kunden

mehr als Vertragsverhandlung

Eine aktive und enge Zusammenarbeit auf Augenhöhe mit dem Kunden, welche kurzfristige Reaktionen auf veränderte Rahmenbedingungen erlaubt, ist wichtiger als sorgsam formulierte Vertragskonstrukte.

Reagieren auf Veränderung **mehr**

als das Befolgen eines Plans

Pläne sind wichtig, müssen aber genug Raum für Anpassungen lassen, da es bei hochkomplexen Projekten und sich stetig ändernden Rahmenbedingungen nahezu unmöglich ist alle Eventualitäten und Ereignisse im Vorfeld zu planen.

12 AGILE PRINZIPIEN

Agile Prinzipien

Neben dem agilen Manifest wurden noch 12 weitere Prinzipien formuliert, an denen sich agile Teams orientieren sollten



Kunden zufriedenstellen

Unsere höchste Priorität ist es, den **Kunden durch frühe und kontinuierliche Auslieferung** wertvoller Software zufrieden zu stellen.



Liefere regelmäßig

Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.



Funktionierende Software

Funktionierende Software ist das wichtigste Fortschrittsmaß.



Erhalte eine konstante Geschwindigkeit

Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.



Unterstütze motivierte Personen

Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.



Nutze "Face-to-Face" Kommunikation

Die effizienteste und effektivste Methode, **Informationen** an und innerhalb eines Entwicklungsteams zu **übermitteln**, ist im **Gespräch von Angesicht zu Angesicht**.

12 AGILE PRINZIPIEN

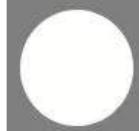
Agile Prinzipien

Neben dem agilen Manifest wurden noch 12 weitere Prinzipien formuliert, an denen sich agile Teams orientieren sollten



Mit Fachexperten zusammenarbeiten

Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.



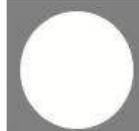
Teams sollten sich selbst organisieren

Die besten Architekturen, Anforderungen und Entwürfe entstehen durch **selbstorganisierte Teams**.



Reflektieren und anpassen

In **regelmäßigen Abständen** reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.



Fokus auf technische Exzellenz

Ständiges Augenmerk auf **technische Exzellenz** und gutes Design fördert Agilität.



Einfachheit

Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.

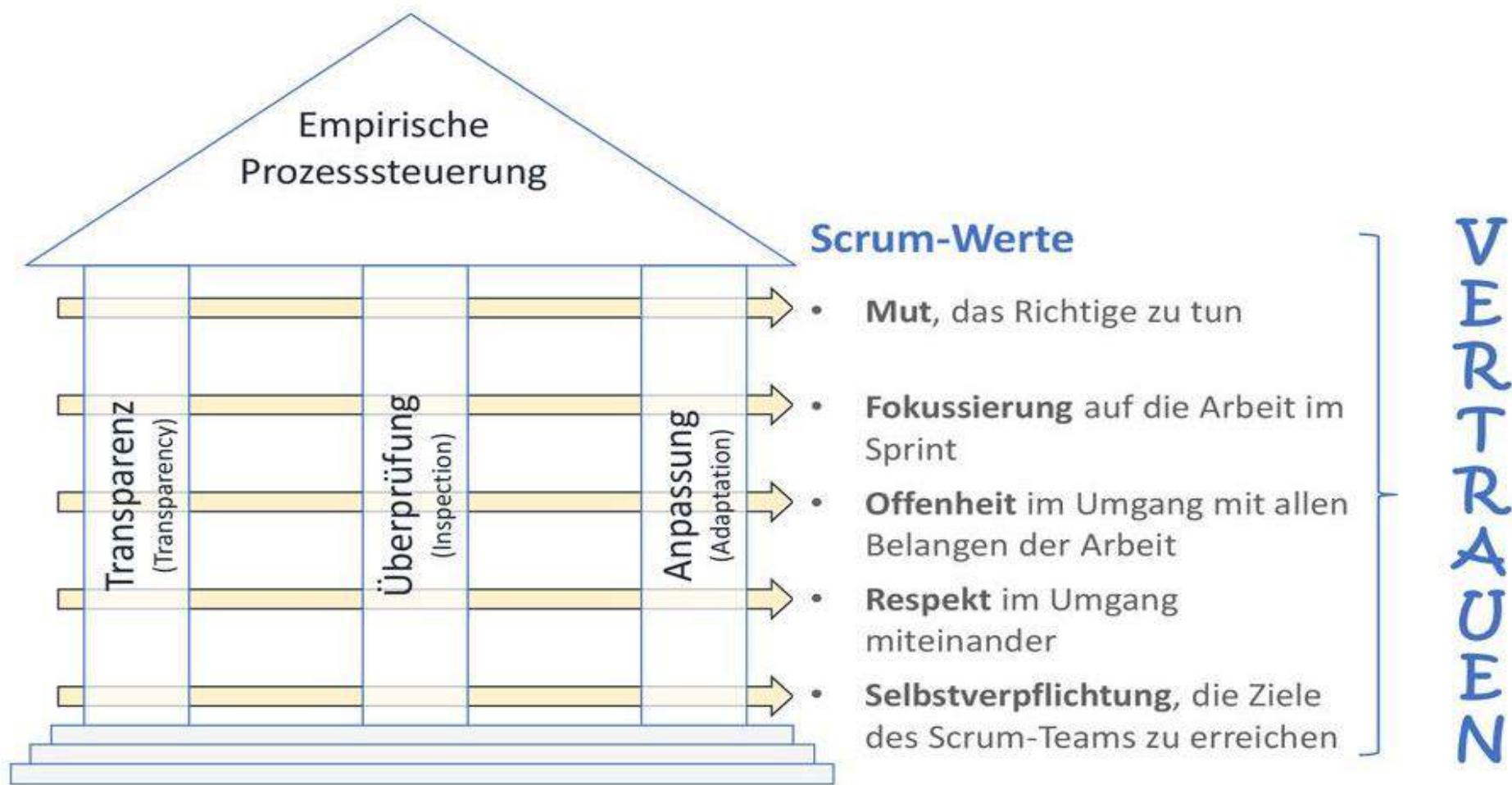


Änderungen sind Willkommen

Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden

SCRUM – SÄULEN UND WERTE

Scrum: Werte – Basis für Vertrauen



5 SCRUM-WERTE



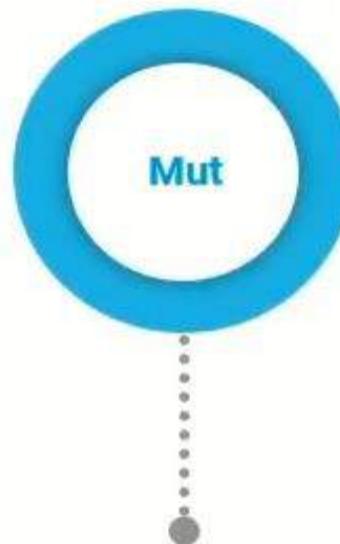
Respekt

Gegenseitiger Respekt ist die Grundlage für eine offene und konstruktive Zusammenarbeit



Fokus

Scrum wird in kurzen Iterationen (Sprints) mit klaren Zielen abgewickelt. Sprintziele lassen sich nur mit einem klaren Fokus erreichen



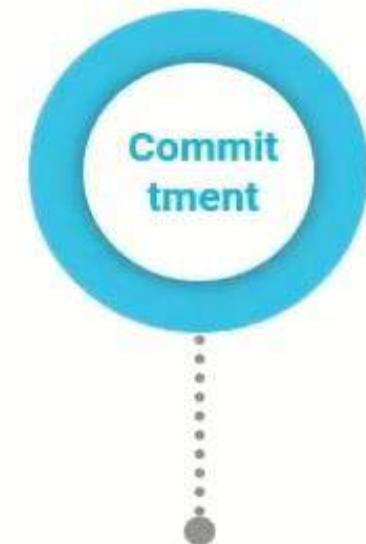
Mut

Scrum Teammitglieder gehen mutig neue Wege und sind Veränderungen gegenüber aufgeschlossen, auch wenn diese mal in Rückschlägen enden



Offenheit

Das Scrum Team und seine Stakeholder kommunizieren offen und ehrlich über den aktuellen Entwicklungsstand, mögliche Verbesserungen und bestehende Probleme



Commitment

Das Team ist intrinsisch motiviert und bekennt sich zum Produkt, den Sprintzielen und gemeinsam getroffenen Vereinbarungen

SCRUM-WERTE

EINE GROSSES POTENTIAL

Umfeldfaktoren

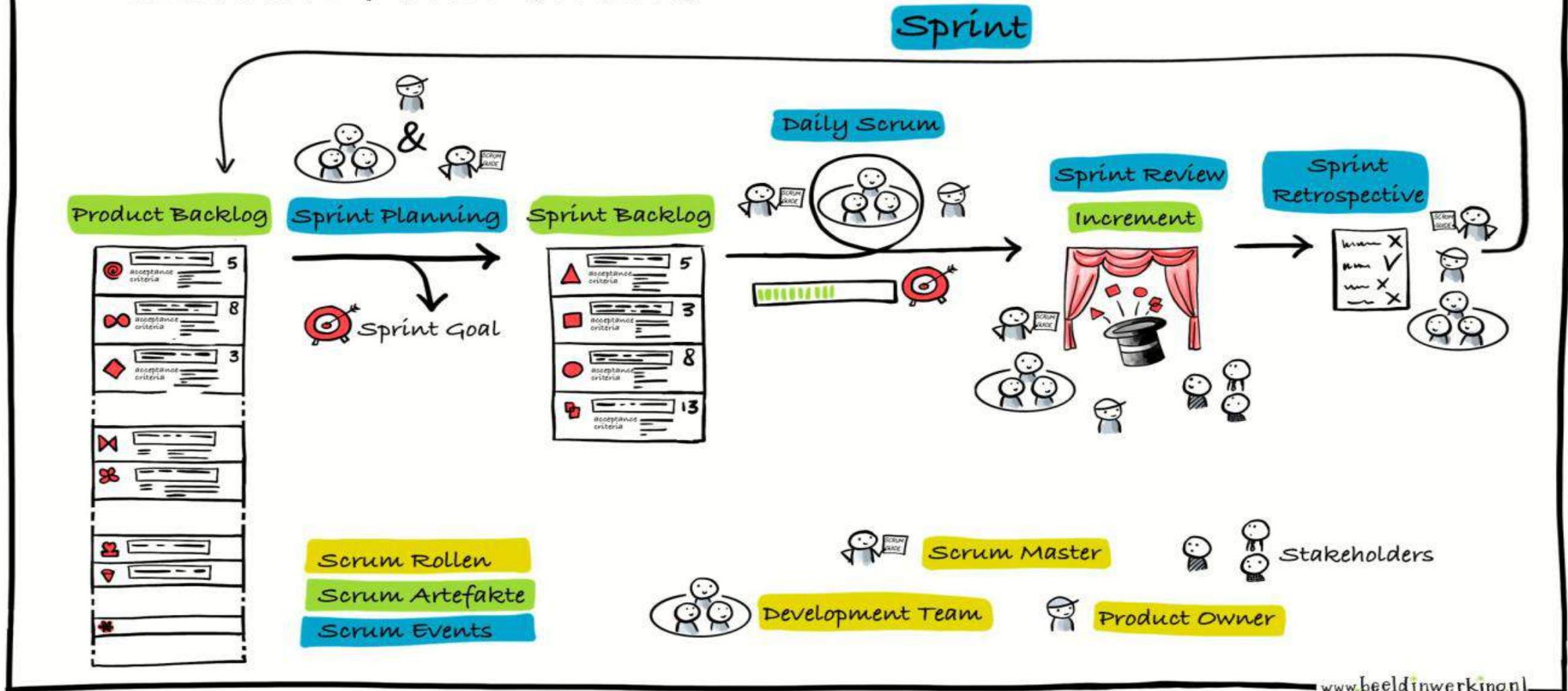
- Grundlagen und hoch-perf. Teams

Hoch-performante Teams bringen erstaunliche Ergebnisse hervor!



PROJEKTMANAGEMENT MIT SCRUM

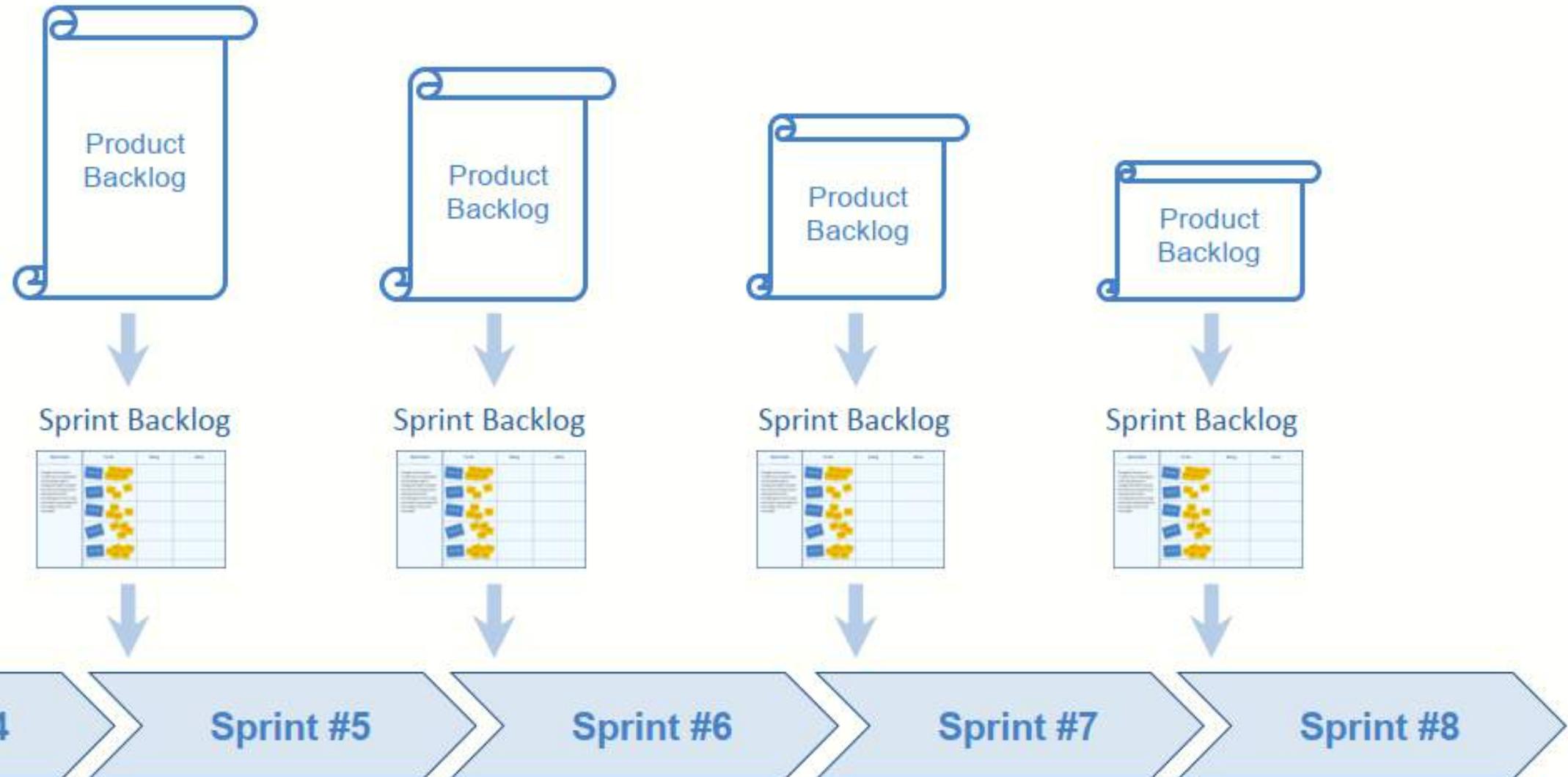
Scrum Framework



PROJEKTMANAGEMENT MIT SCRUM

Scrum Rollen	Scrum Artefakte	Scrum Events
<p>Scrum Master:</p> <ul style="list-style-type: none">▶ Hilft dem Scrum Team & der Organisation die bestmögliche Produktivität zu erreichen▶ Coacht Scrum nach dem Scrum Guide und schaut, dass die Scrum Elemente verstanden und gelebt werden <p>Product Owner:</p> <ul style="list-style-type: none">▶ Trägt die Verantwortung über das Produkt und damit über das Product Backlog▶ Legt fest, welche Produktneuerungen in welcher Reihenfolge an den Kunden ausgeliefert werden▶ Optimiert den Wert der bevorstehenden Arbeit <p>Das Development Team:</p> <ul style="list-style-type: none">▶ Ist für die Umsetzung der Produktänderungen verantwortlich▶ Selbstorganisiert und cross-funktional setzt es sie um und kann sie an den Kunden ausliefern▶ Stellt in jedem Sprint ein auslieferbares Product Increment zur Verfügung <p>Alle gemeinsam bilden das Scrum Team.</p>	<p>Product Backlog:</p> <ul style="list-style-type: none">▶ Liste aller noch offenen Neuerungen, die am Produkt durchgeführt werden sollen▶ Das Product Backlog ist so sortiert, wie es der Product Owner für richtig hält <p>Sprint Backlog:</p> <ul style="list-style-type: none">▶ Liste aller Aufgaben, die vom Development Team in diesem Sprint umgesetzt werden <p>Product Increment:</p> <ul style="list-style-type: none">▶ Das neueste Produkt▶ Umfasst die Summe aller Änderungen, die bis heute umgesetzt wurden	<p>Sprint:</p> <ul style="list-style-type: none">▶ Ein Container über alle weiteren Scrum Events▶ Im Sprint wiederholen sich die weiteren Scrum Events in der angegebenen Reihenfolge <p>Sprint Planning:</p> <ul style="list-style-type: none">▶ Das Sprint Ziel (Sprint Goal) wird festgelegt▶ Es wird geplant, welche Produktänderungen in diesem Sprint umgesetzt werden <p>Daily Scrum:</p> <ul style="list-style-type: none">▶ Das Development Team inspiriert und plant seinen täglichen Fortschritt zum Sprint Goal <p>Sprint Review:</p> <ul style="list-style-type: none">▶ Scrum Team & Stakeholder schauen sich das neue Product Increment an, also die Produktänderungen, die nun an den Kunden ausgeliefert werden können <p>Sprint Retrospektive:</p> <ul style="list-style-type: none">▶ Das Scrum Team inspiriert den Sprint: Was ist gut gelaufen? Wo können wir uns verbessern?▶ Gemeinsam einigt man sich auf eine Verbesserung z.B. eines Prozesses, die im kommenden Sprint umgesetzt wird

WIE FUNKTIONIERT SCRUM?



Sprint #4

Sprint #5

Sprint #6

Sprint #7

Sprint #8

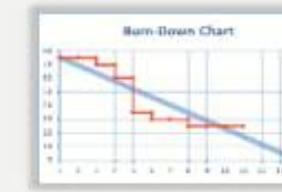
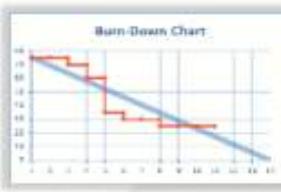
↓
Increment #4

↓
Increment #5

↓
Increment #6

↓
Increment #7

“Done”, and potentially releasable



Feedback

Feedback

Feedback

Feedback

To be applied to the Product Backlog

Sprint Planning

Daily Scrum

Sprint Review

Sprint Retrospective

Sprint #6

WAS IST SCRUM?

DER ABLAUF

Ablauf

Viele kurze **Iterationen** (Sprints) an deren Ende stets ein **“Ergebnis”** steht

Fokus auf jene Anforderungen die **aktuell** die **höchste Priorität** haben, bzw. Den höchsten **Nutzen** (business value) versprechen



WAS IST SCRUM?

Prinzipien

Kurze
Iterationen

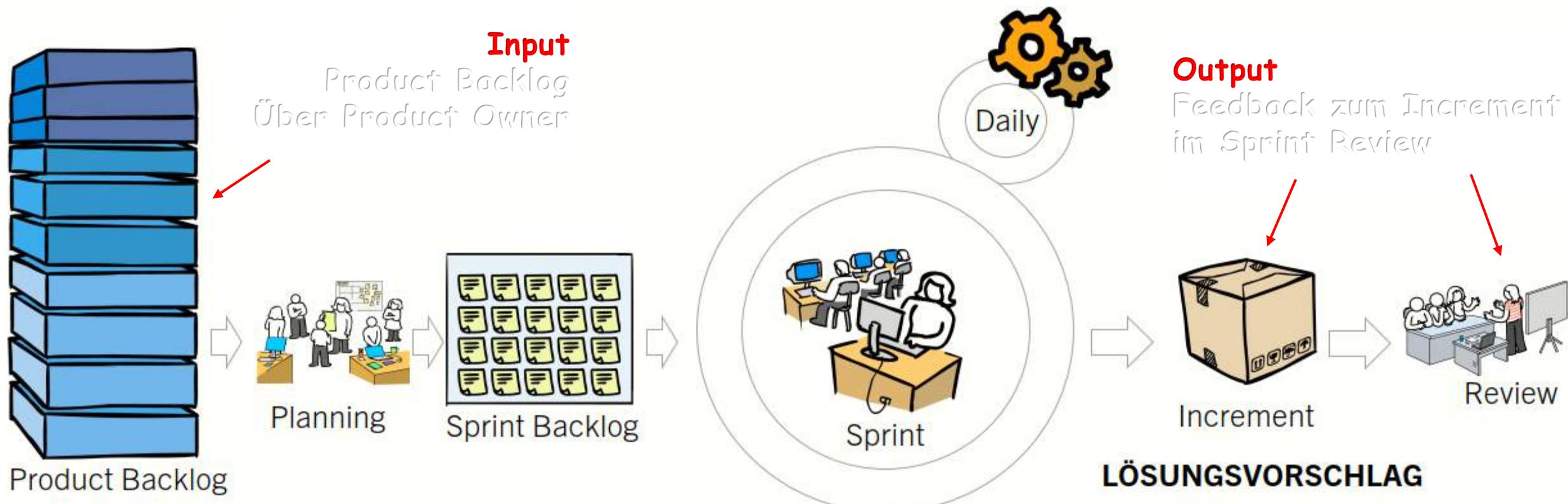
Selbst-
organisierende
Teams

Kontinuierliche
Verbesserung

Transparenz

SCRUM BEDIENEN

ZWEI SCHNITTSTELLEN: INPUT UND OUTPUT



WAS?
PRIORITÄT

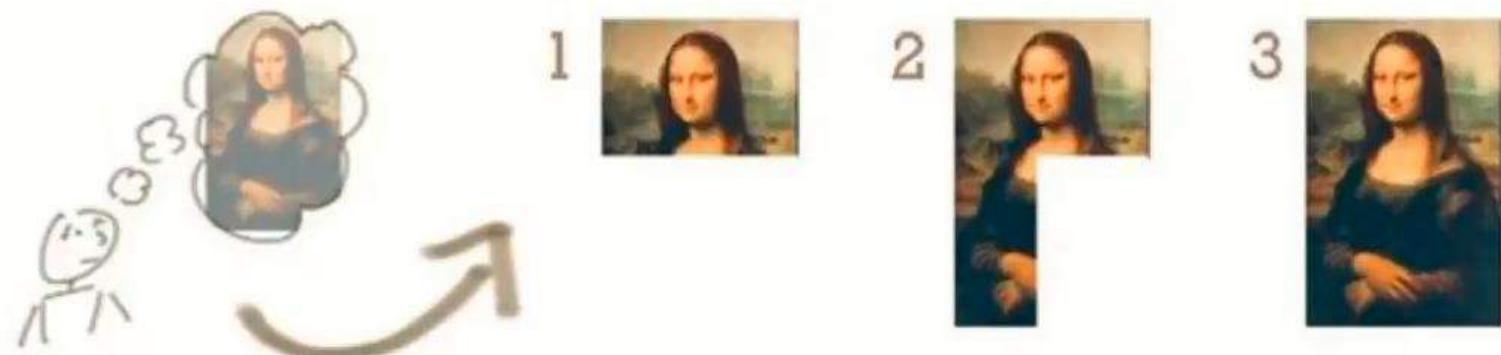
FEEDBACK

VERGLEICH INKREMENTELLER UND ITERATIVER ANSATZ

Inkrementell versus Iterativ

Es ist wichtig, den Unterschied zwischen inkrementellen und iterativen Ansätzen unterscheiden zu können

Inkrementeller Ansatz



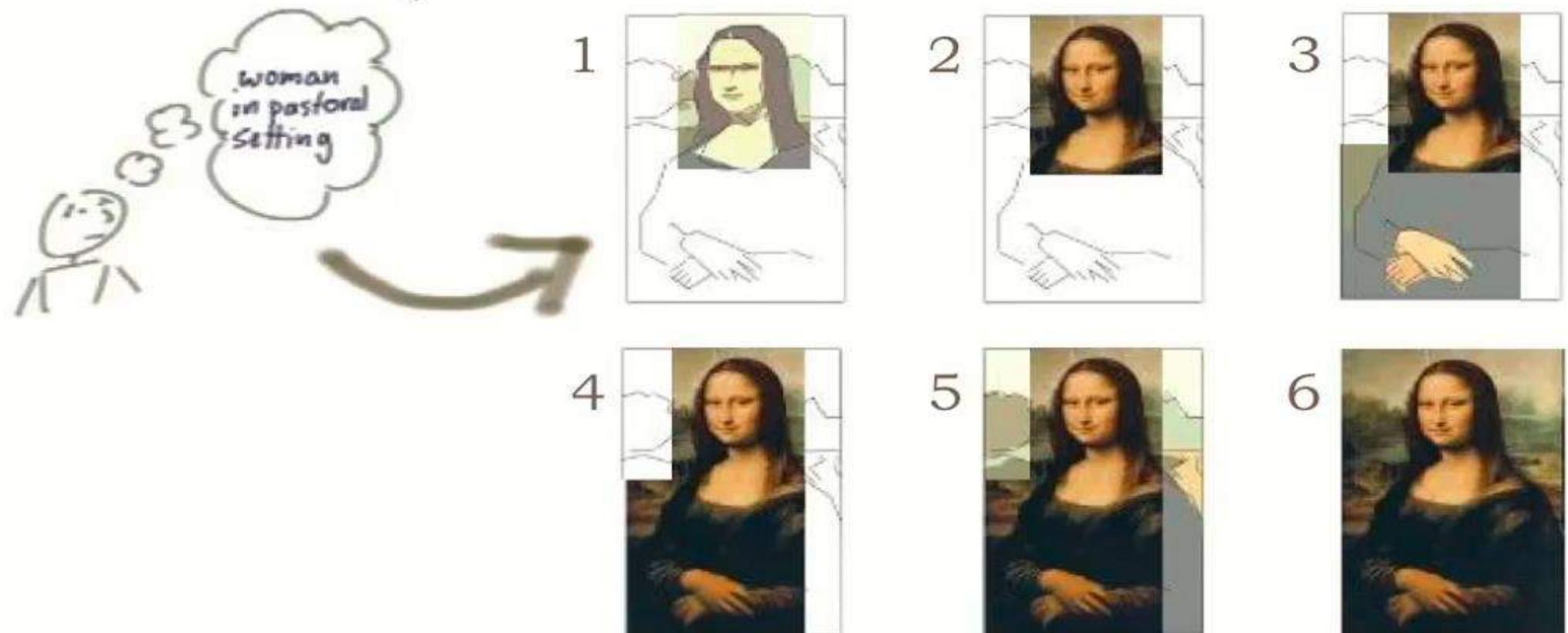
Iterativer Ansatz



VERGLEICH INKREMENTELLER UND ITERATIVER ANSATZ

Scrum verbindet beide Ansätze und strebt so eine kontinuierliche Weiterentwicklung und Verbesserung des jeweiligen Produktes an

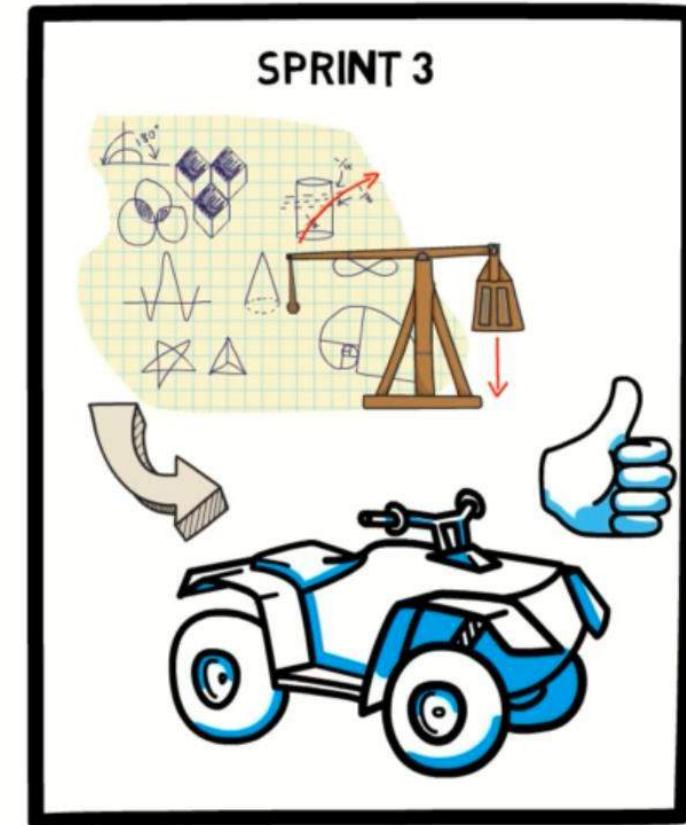
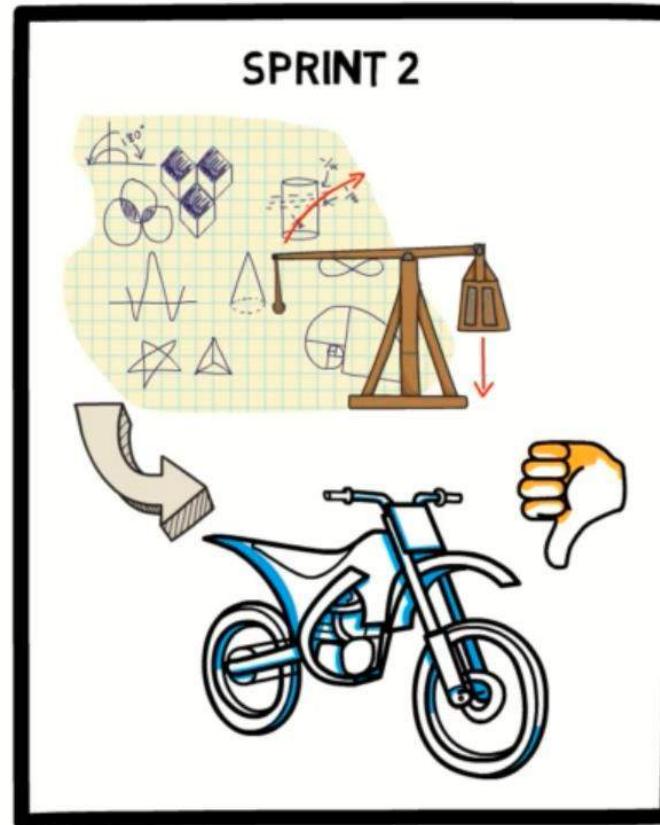
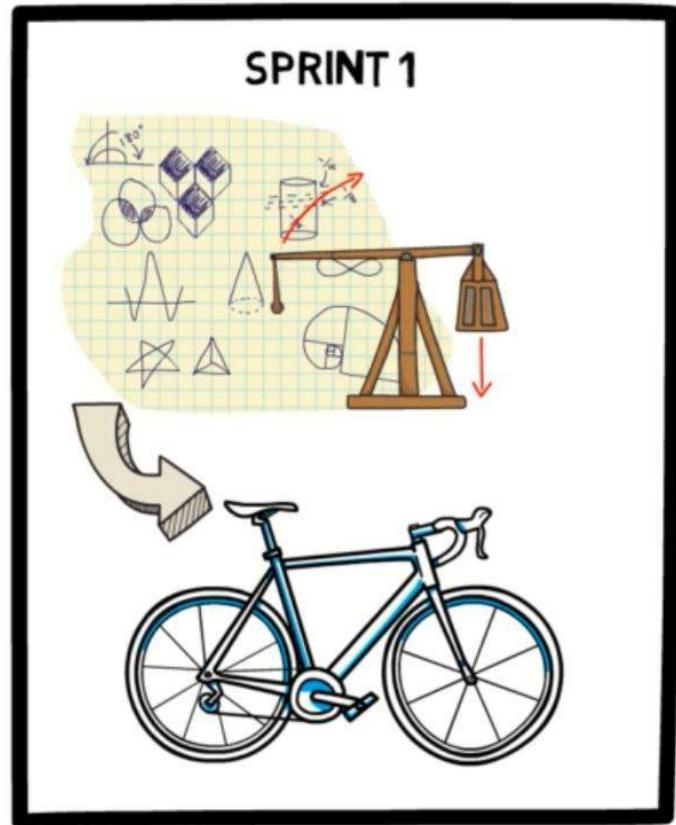
Inkrementeller und iterativer Ansatz



VERGLEICH INKREMENTELLER UND ITERATIVER ANSATZ

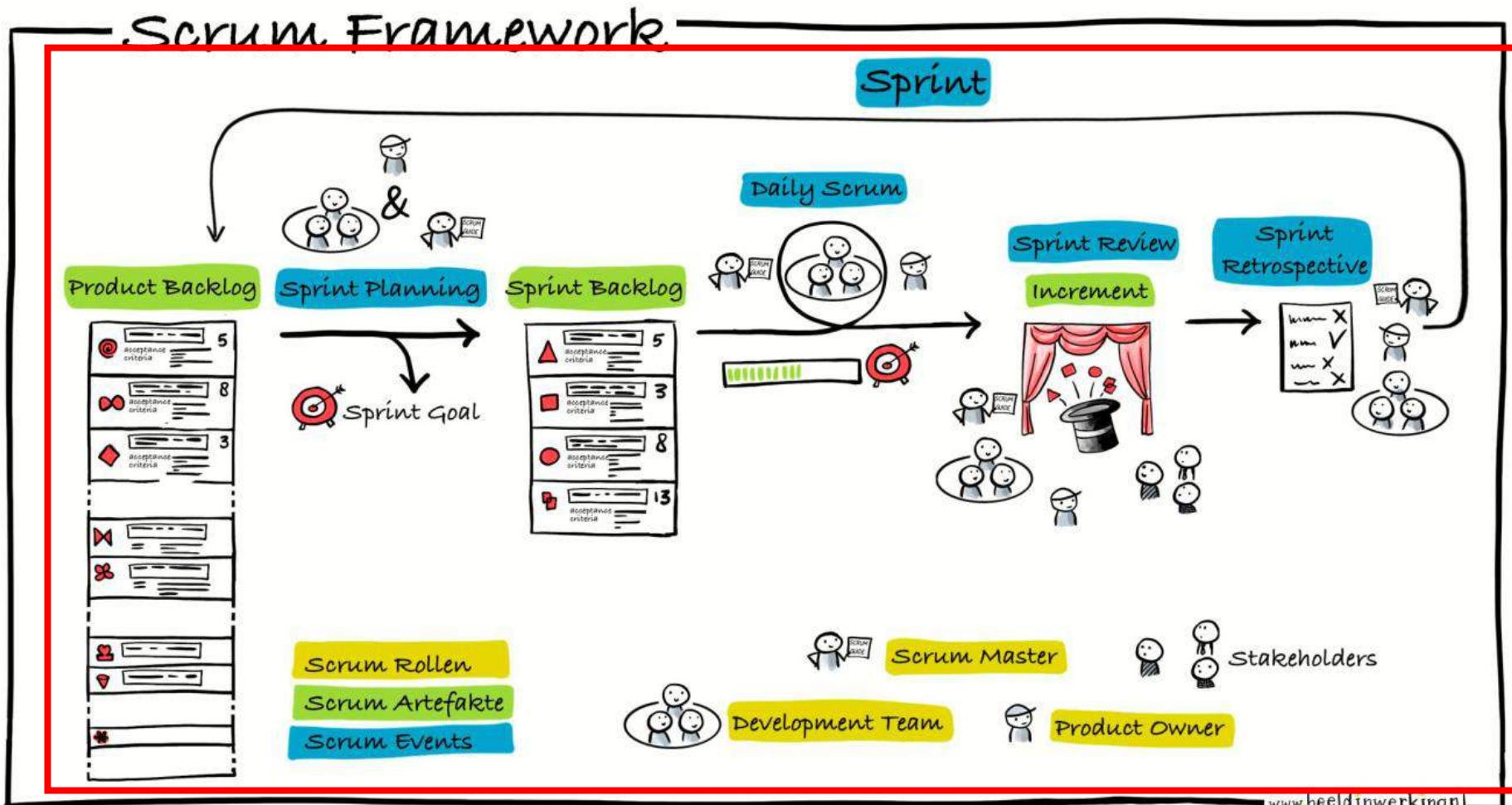


WIE FUNKTIONIERT SCRUM?

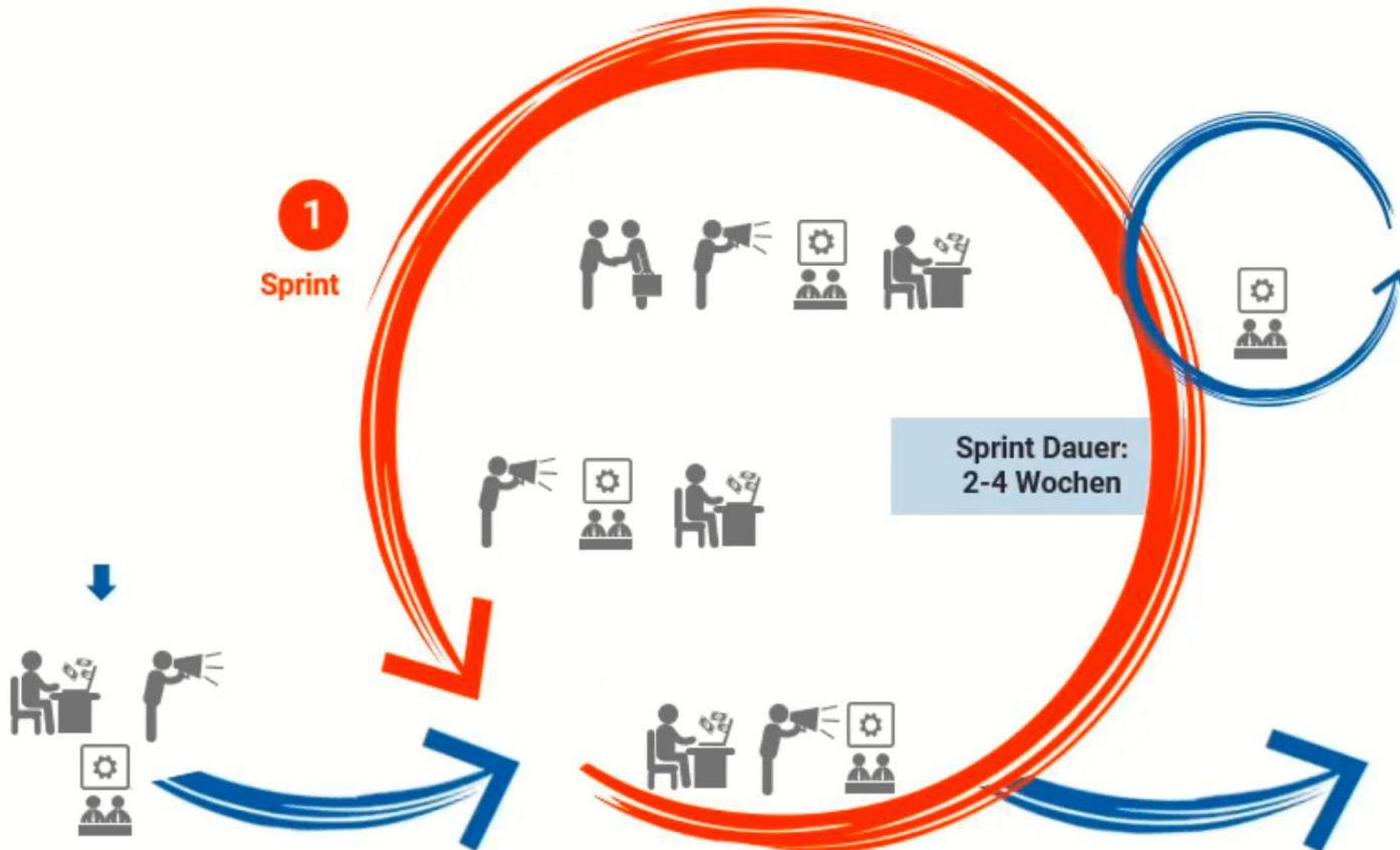


Das Scrum Team strebt in jedem Sprint die Entwicklung eines fertigen, getesteten und auslieferbaren Produkt Inkrementen an

SCRUM EVENTS SPRINT

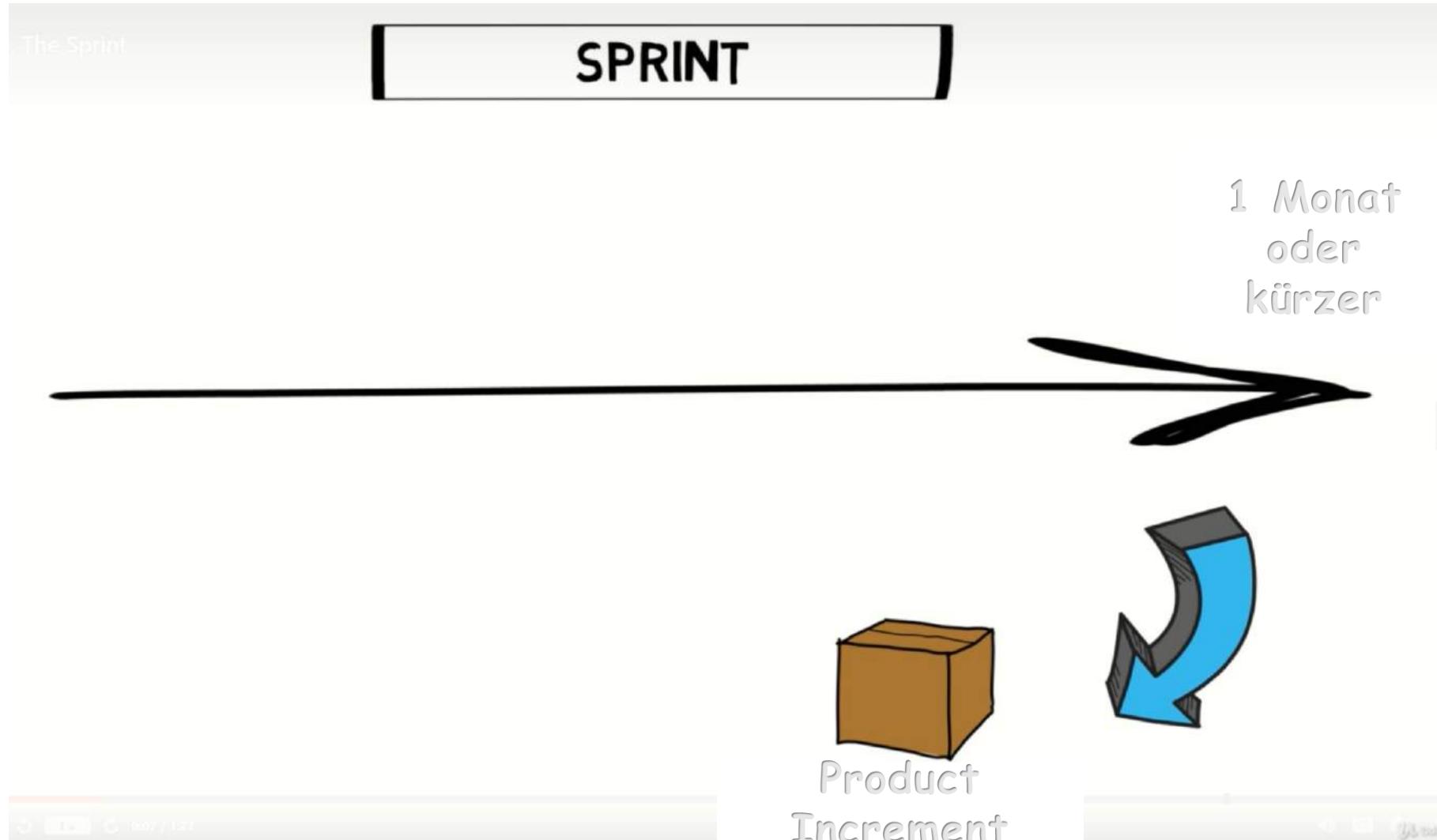


SCRUM EVENTS: SPRINT



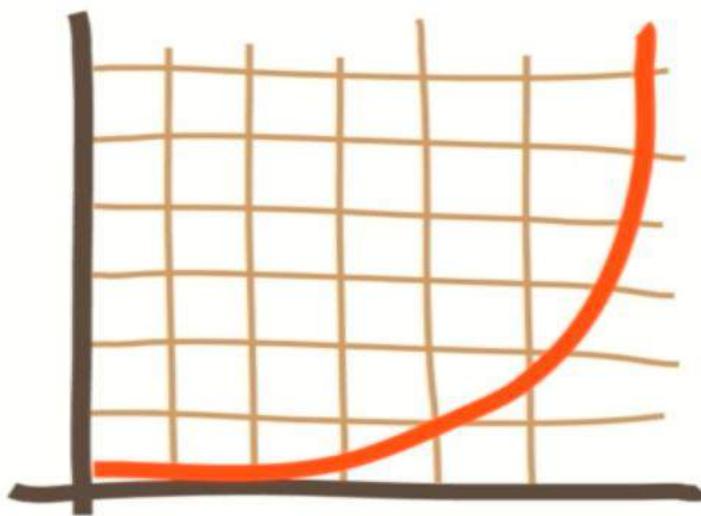
SCRUM EVENTS

SPRINT



SCRUM EVENTS SPRINT

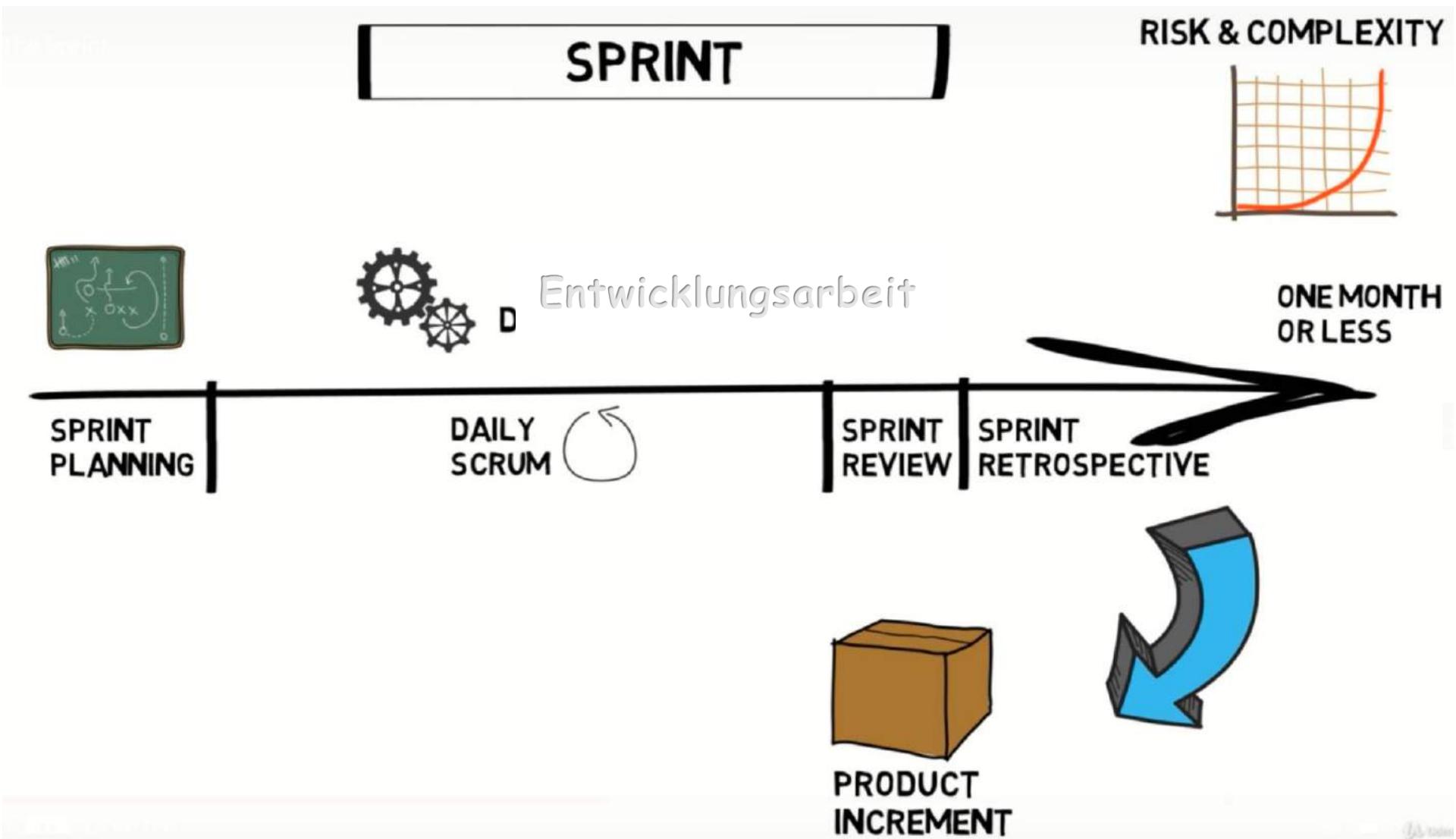
RISK & COMPLEXITY



**ONE MONTH
OR LESS**

SCRUM EVENTS

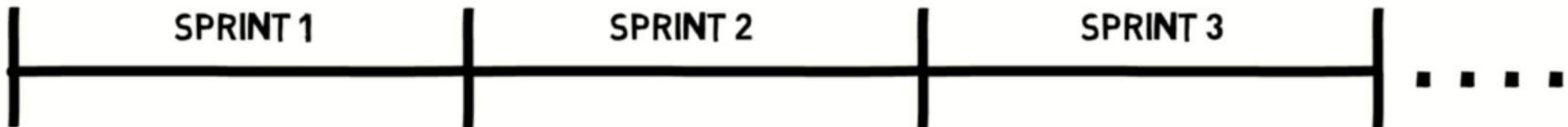
SPRINT



SCRUM EVENTS

SPRINT

Der Folge-Sprint fängt **IMMER** an, sobald
der Vorgänger-Sprint beendet ist
-> keine Pause oder Zwischenaktivitäten



SCRUM EVENTS

SPRINT

Abbruch eines Sprints

- Tritt äußerst selten ein, wenn...
 - Der Sprint obsolet wird
- Die Organisation sich strategisch neu ausrichtet
- Nur die Product Owner hat die Autorität, einen Sprint abzubrechen

ZUSAMMENFASSUNG SCRUM EVENTS

SPRINT

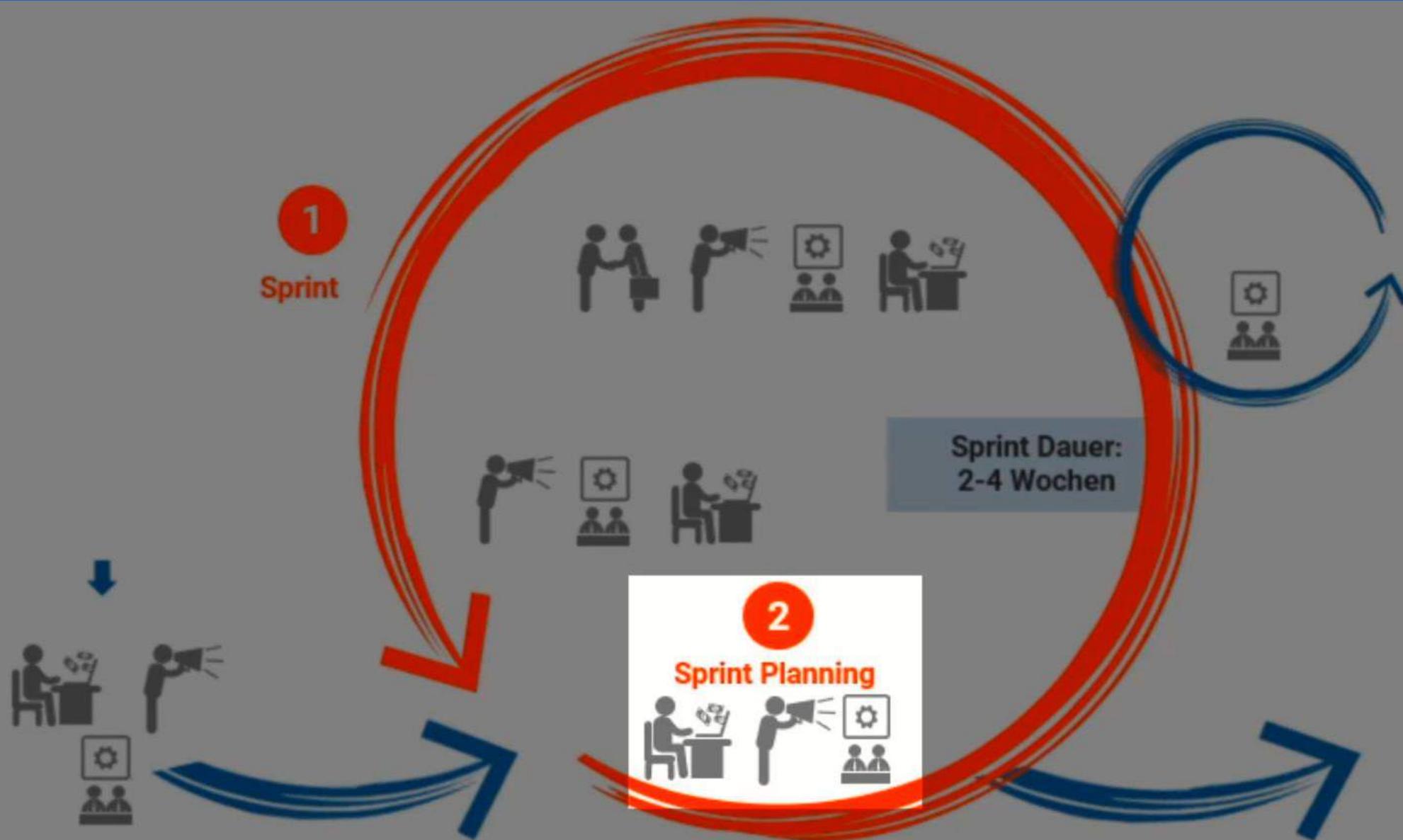
Sprint

Sprint

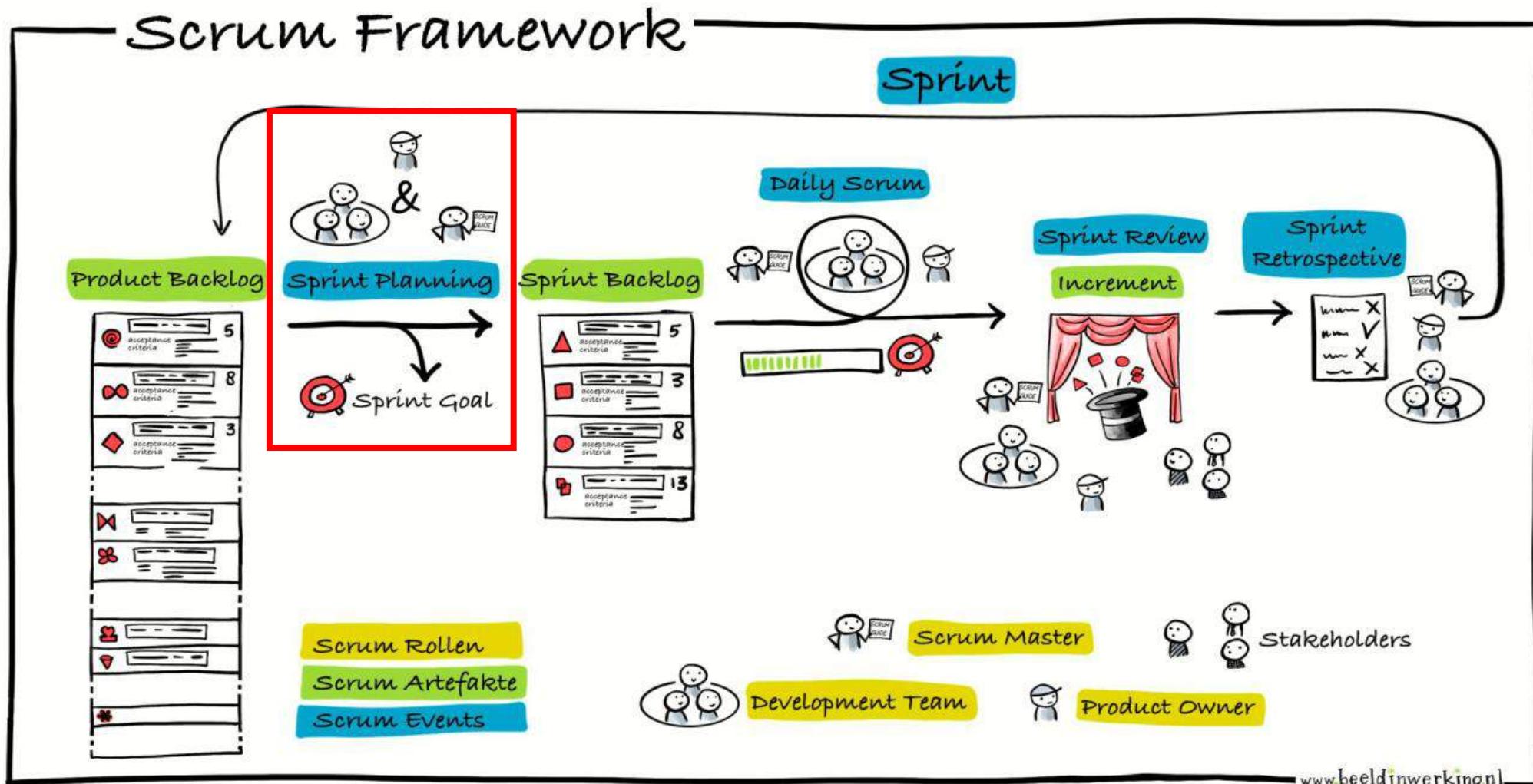
- Time-box für Entwicklungstätigkeiten (Planung, Design, Build, Test, ggfs. Rollout)
- Meist 2 oder 4 Wochen lang
- Umfasst: Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective
- Sprint endet ausschließlich nach Ablauf der Time-box. Ausnahme: Abbruch durch den Product Owner, wenn Sprintziel obsolet

1

SCRUM EVENTS: SPRINT PLANNING

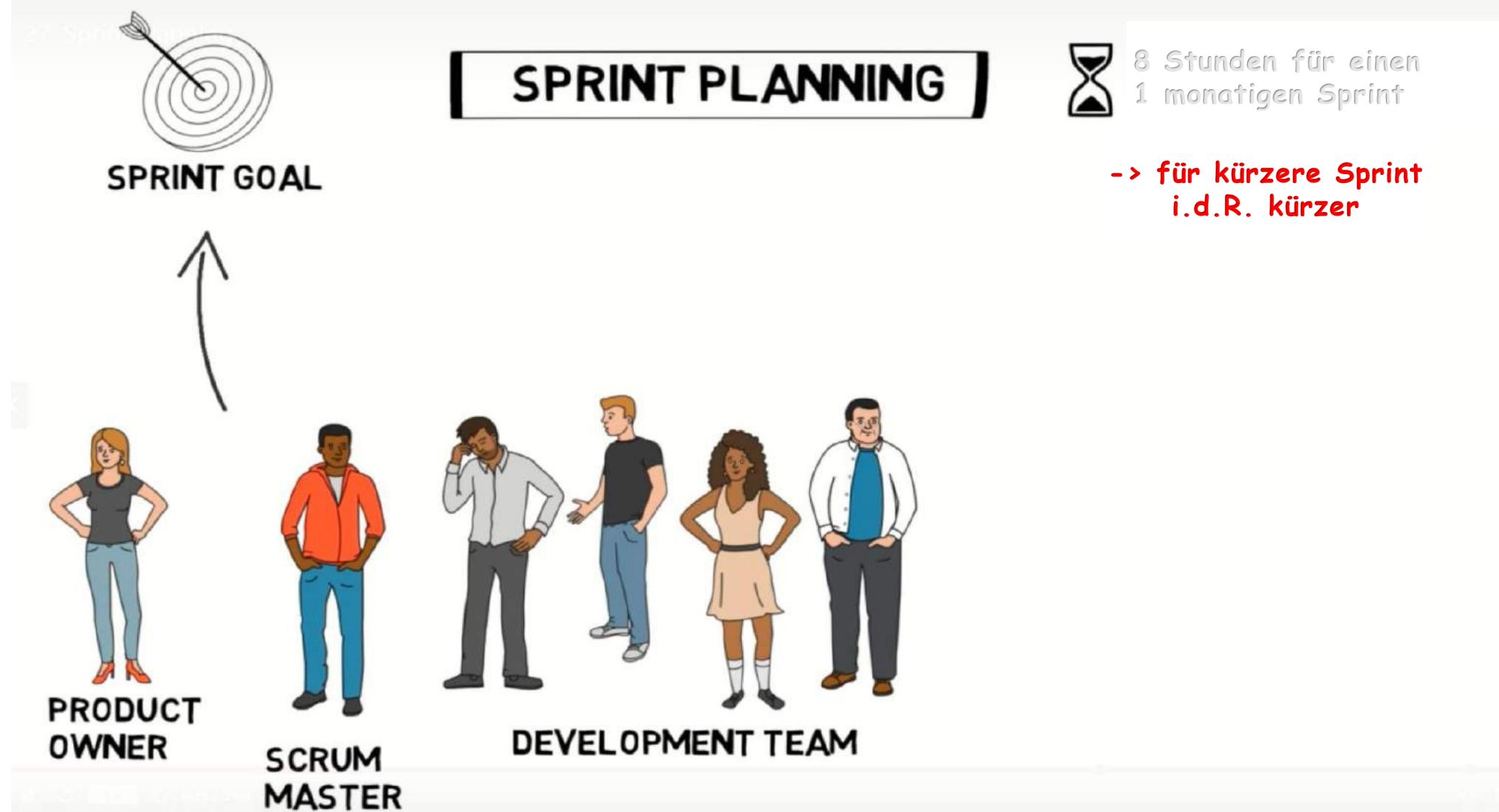


SCRUM EVENTS SPRINT PLANNING



SCRUM EVENTS

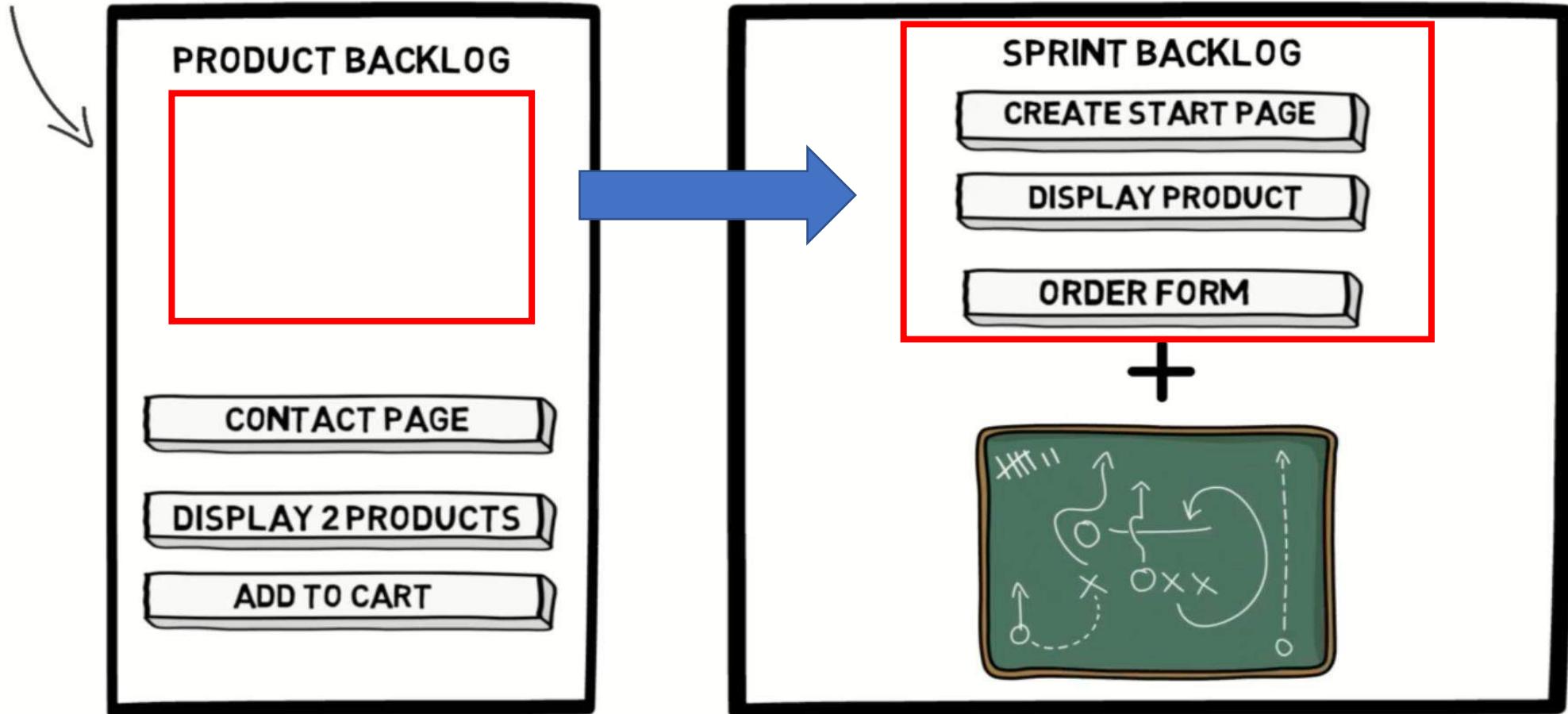
SPRINT PLANNING



SCRUM EVENTS

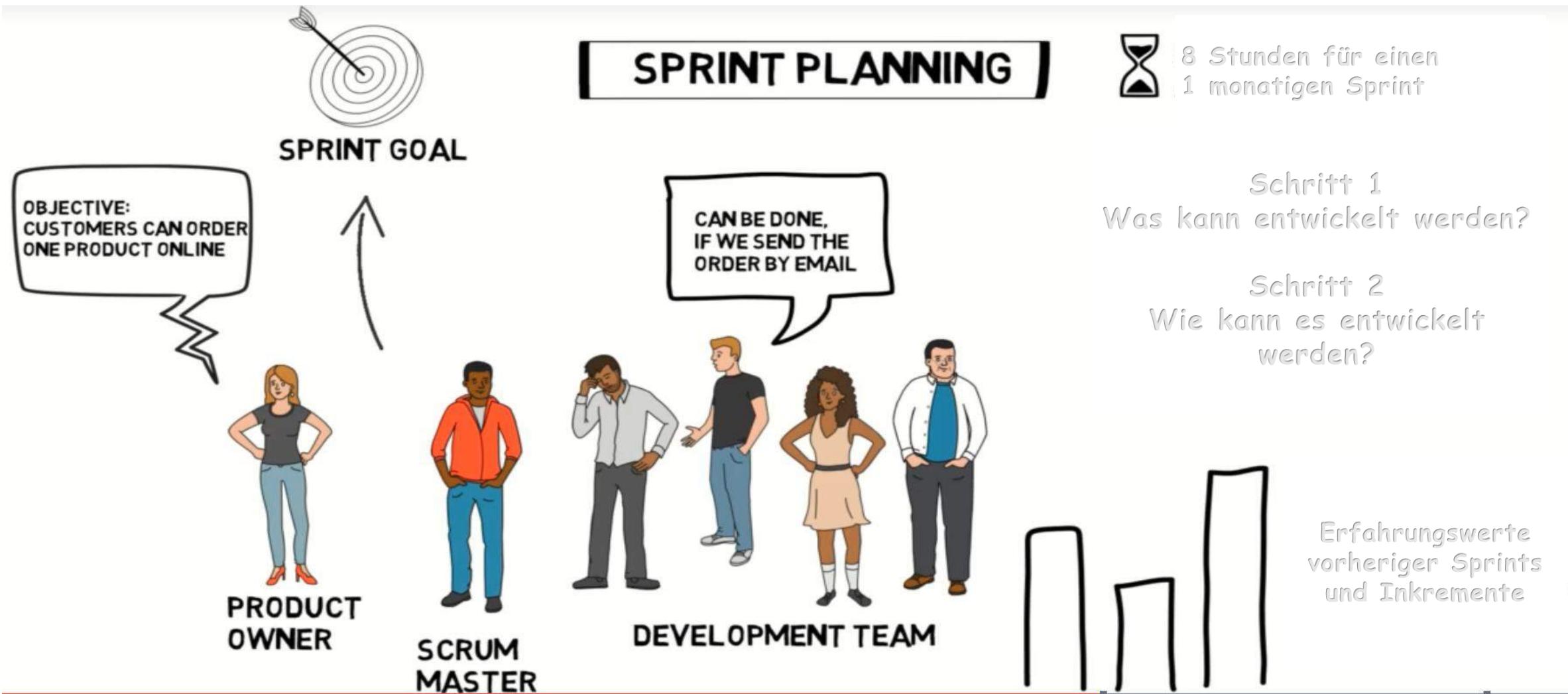
SPRINT PLANNING

PRODUCT BACKLOG ITEMS



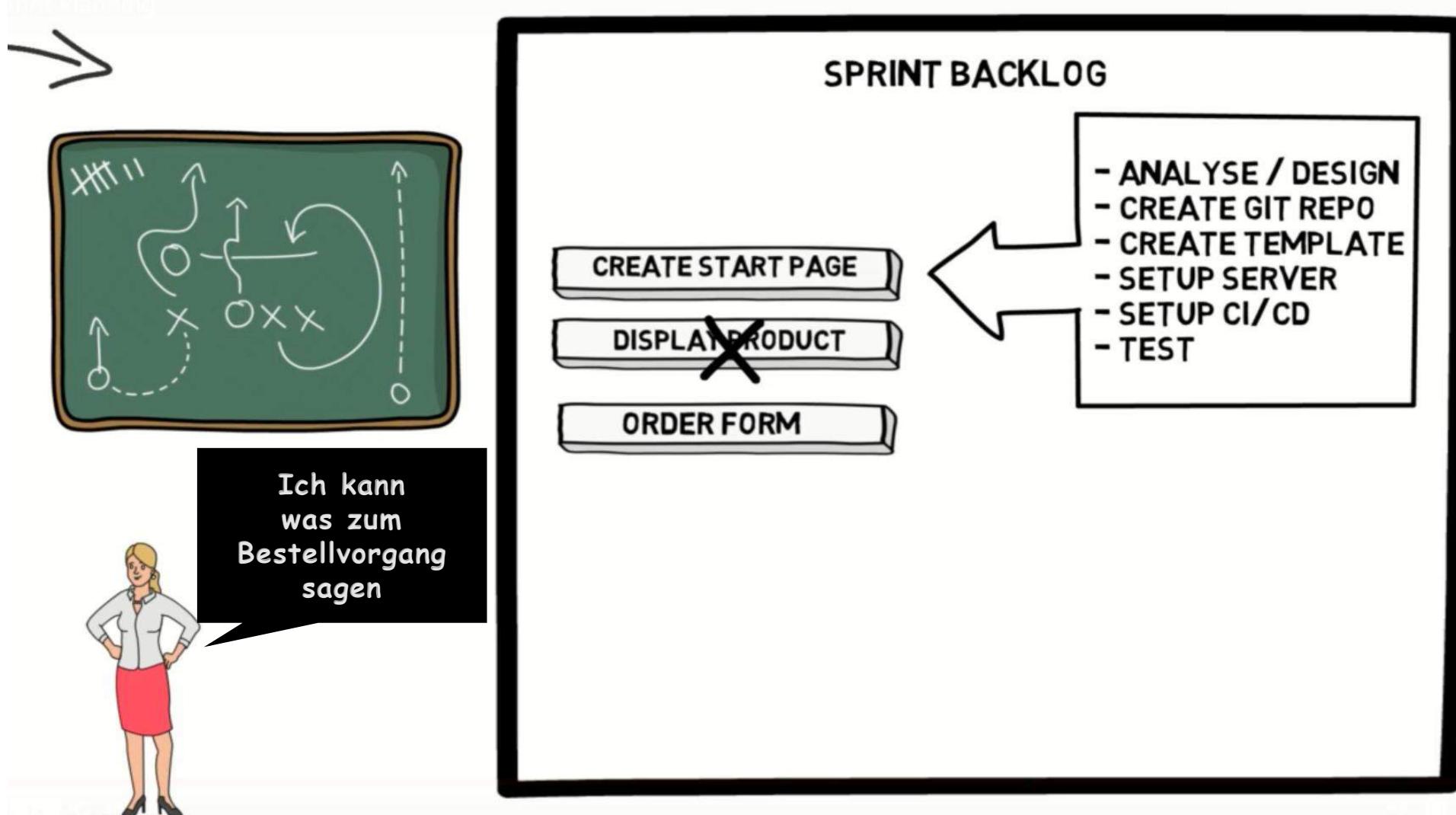
SCRUM EVENTS

SPRINT PLANNING



SCRUM EVENTS

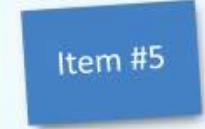
SPRINT PLANNING



SPRINT PLANNING: BEISPIEL AM ENDE DES PLANNINGS

Sprint Goal	To Do	Doing	Done
The goal of this sprint is to make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.	<p>The board illustrates the state of tasks across five items:</p> <ul style="list-style-type: none">Item #1: Contains tasks t.1.6, t.1.3, t.1.2, t.1.4, t.1.1, and t.1.5.Item #2: Contains tasks t.2.1, t.2.2, and t.2.3.Item #3: Contains tasks t.3.4 and t.3.1.Item #4: Contains no visible tasks.Item #5: Contains no visible tasks.		

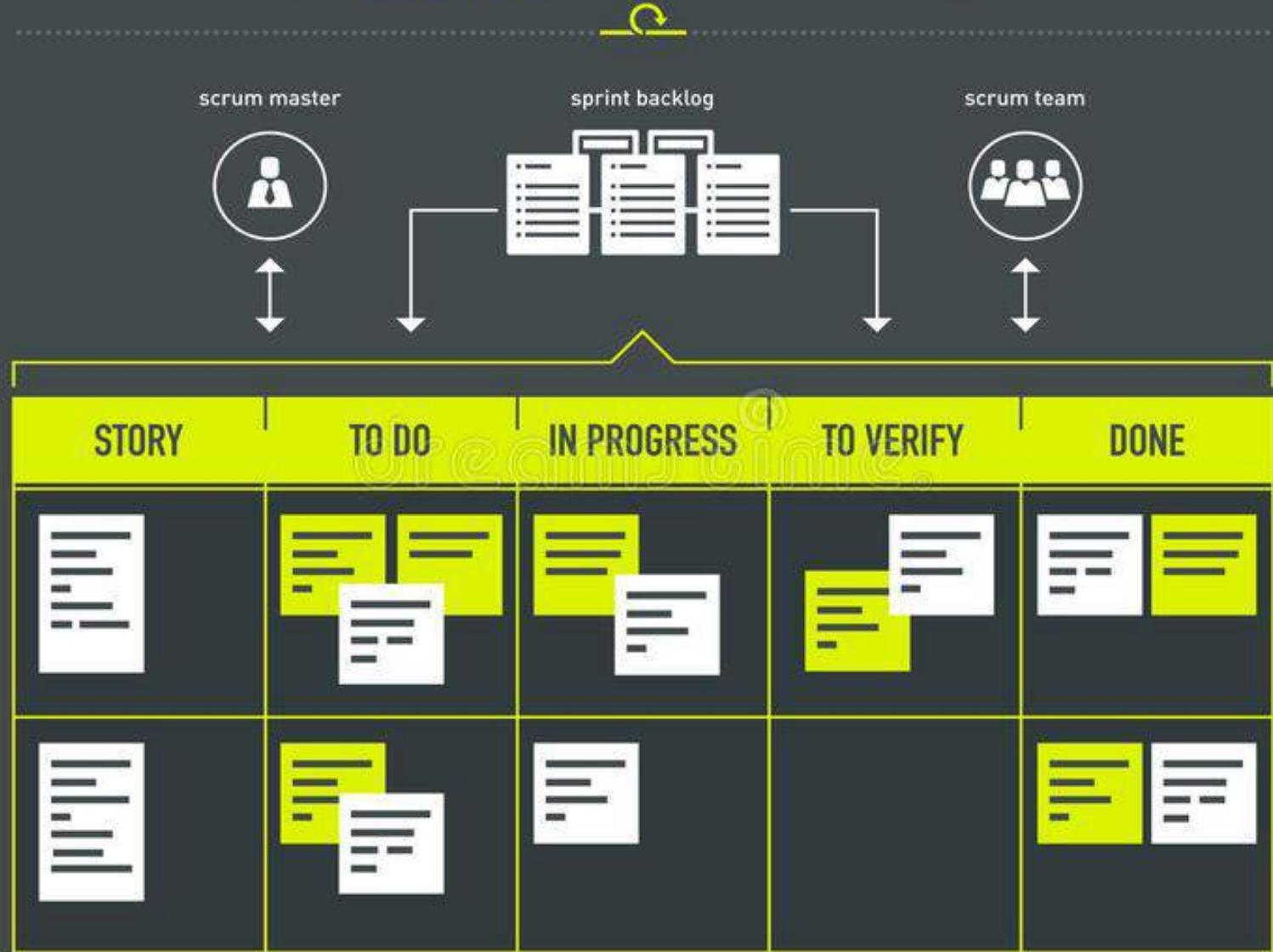
SPRINT PLANNING: BEISPIEL WÄHREND DES SPRINTS

Sprint Goal	To Do	Doing	Done
The goal of this sprint is to make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.			
			
			
			
			

DAS TASK BOARD

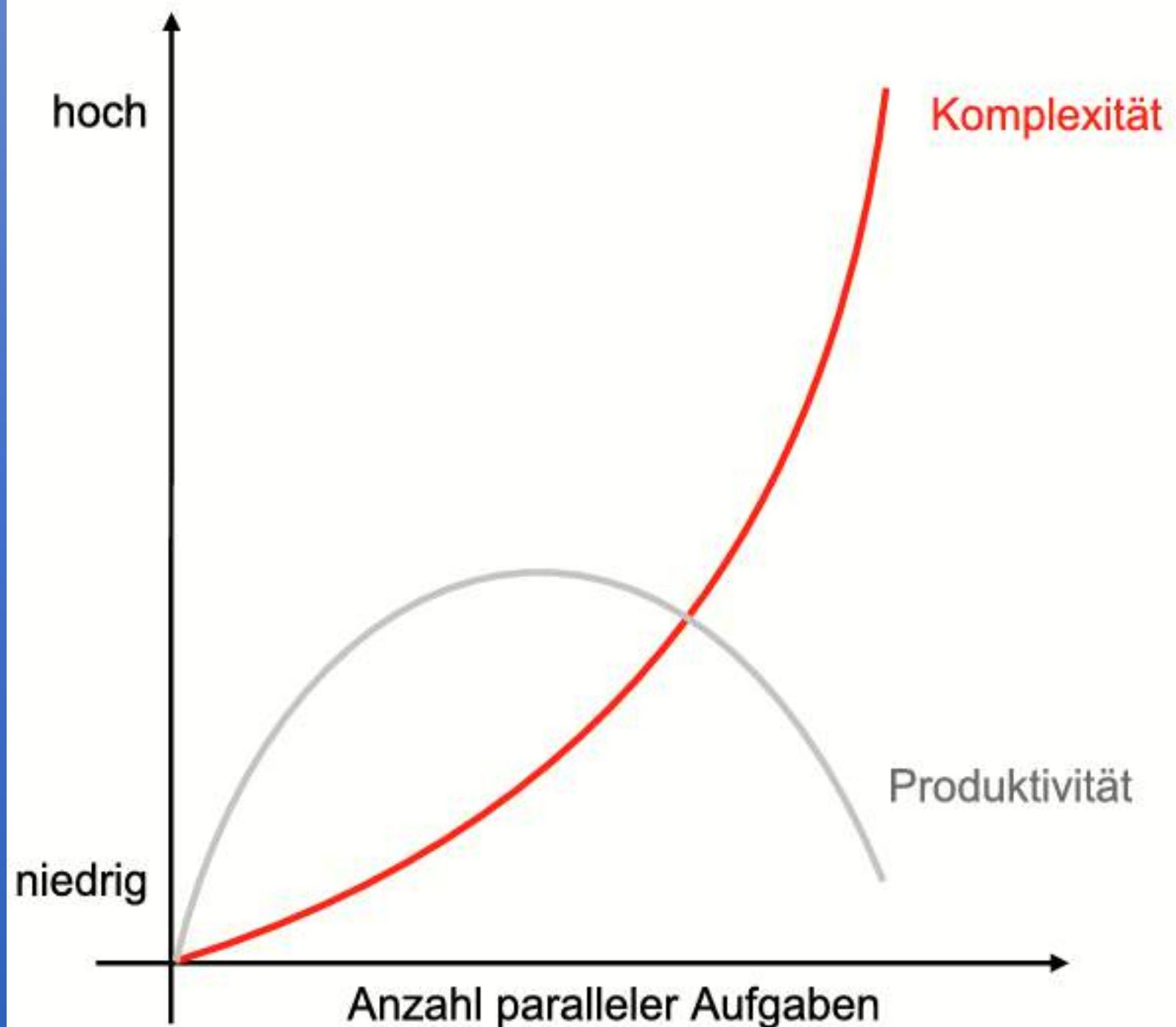
AUFGABEN
TRANSPARENT
VISUALISIEREN

SCRUM TASK BOARD



PARALLELE ARBEIT
BEGRENZEN

WORK-IN-PROGRESS
(WIP)
LIMITS



PARALLELE ARBEIT BEGRENZEN WORK-IN-PROGRESS (WIP) LIMITS

Beispiel:



Herr Bauer hat bereits mehrere parallele Aufgaben, die ihn insgesamt zu 60 % auslasten. Nun bekommt er zusätzlich eine neue Aufgabe übertragen. Er stellt fest, dass ihn die Summe seiner Aufgaben an den Rand seiner Belastungsgrenze bringt, was er sich nicht erklären kann, da rein rechnerisch nur 20 % mehr Auslastung dazugekommen ist. Er weiß nicht, was er zuerst erledigen soll, er vergisst Termine und ihm unterlaufen Fehler, was gar nicht zu ihm passt.

ZUSAMMENFASSUNG SCRUM EVENTS

SPRINT PLANNING

Sprint Planning

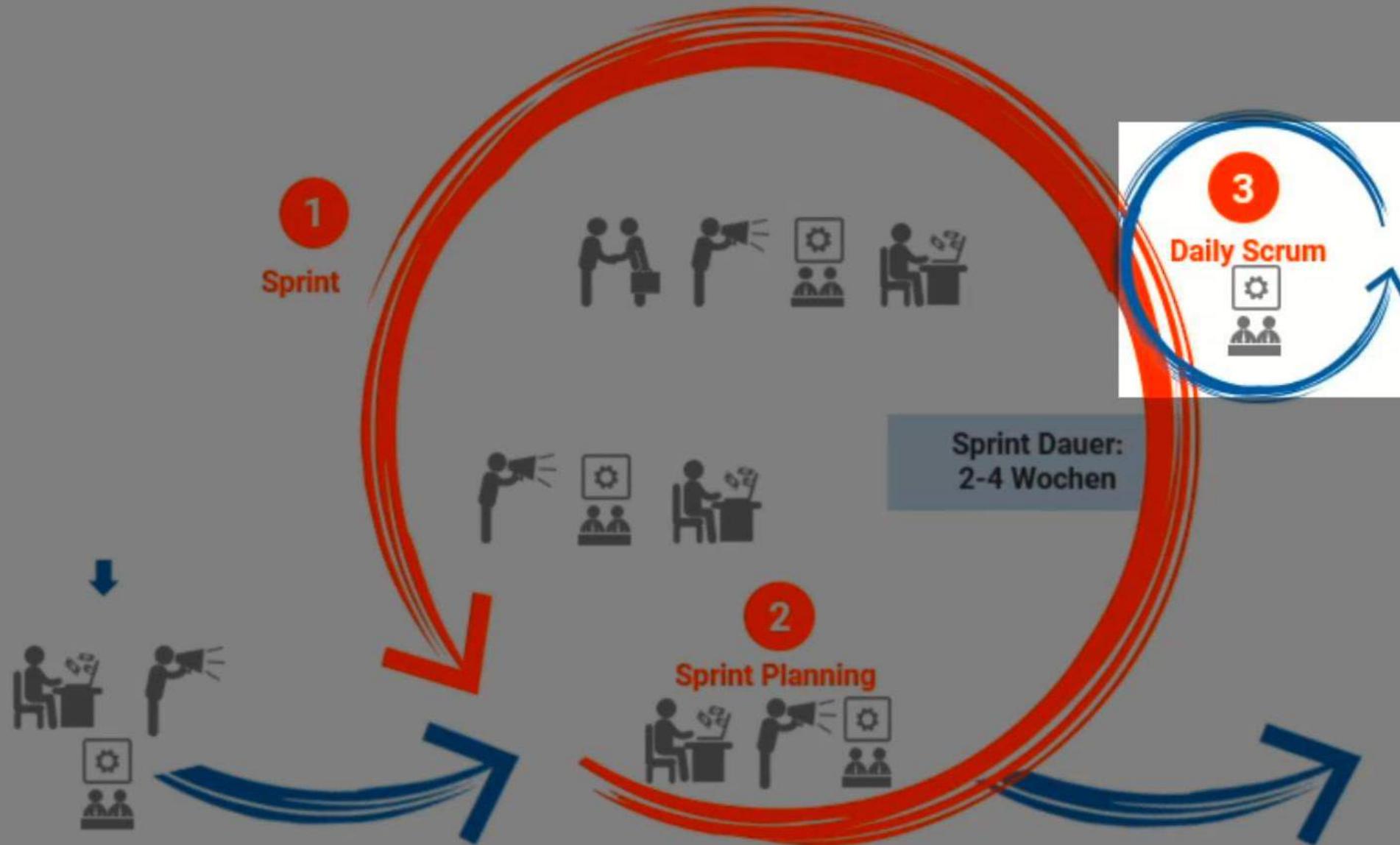
2

Sprint Planning Meeting

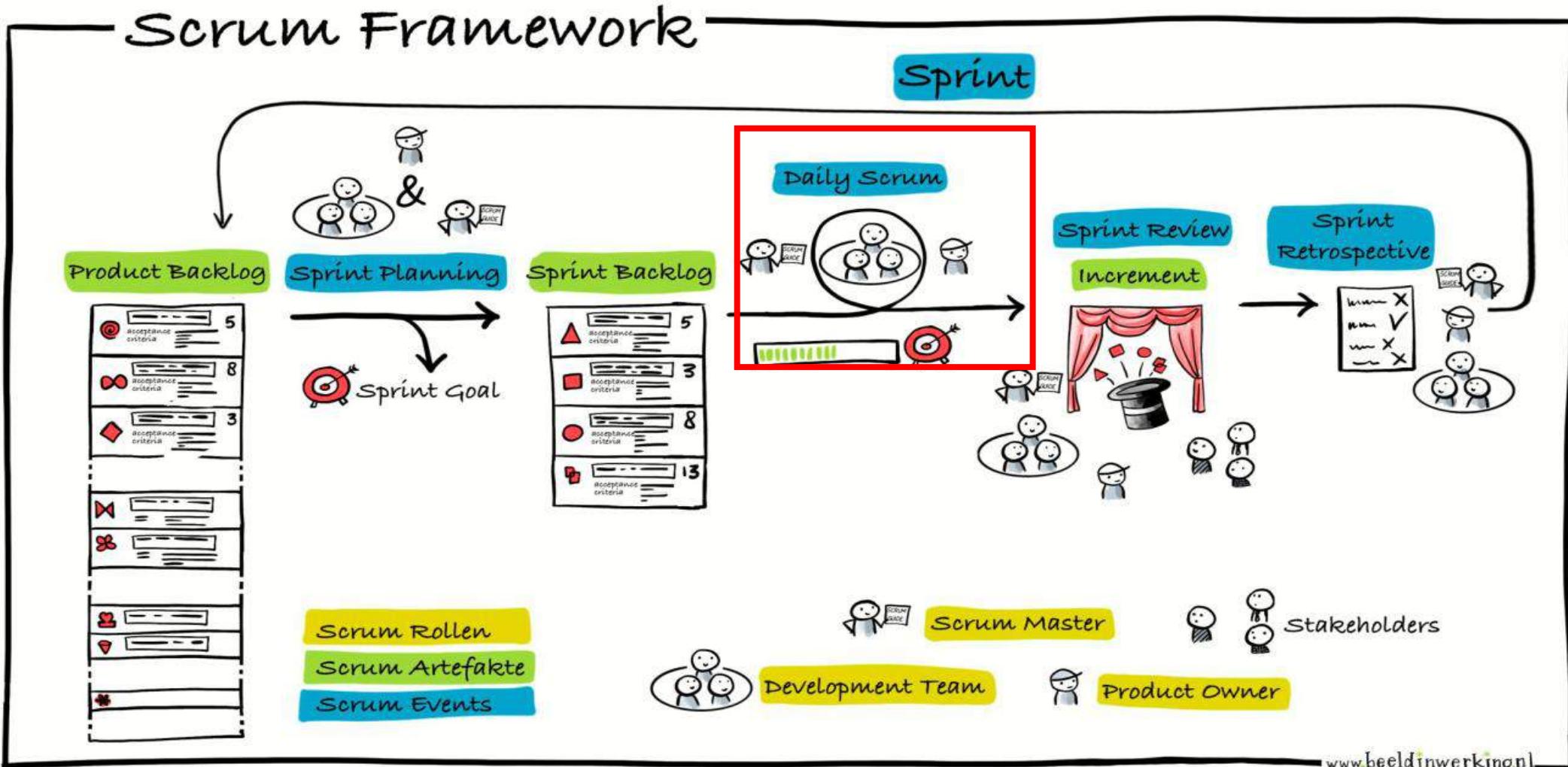
- max. 8 Std. für einen einmonatig. Sprint
- Product Owner, Scrum Master, Entwicklungsteam
- Benötigt ein gepflegtes Product Backlog
- Erstellt das Sprint Backlog, welches die Tätigkeiten für den nächsten Sprint auflistet

Der Product Owner stellt die Anforderungen mit der höchsten Priorität vor und kommuniziert ein Sprintziel, welches mit dem Entwicklungsteam diskutiert wird. Entwicklungsteam und Product Owner legen dann gemeinsam die Anforderungen fest, welche in den nächsten Sprint einfließen. Im weiteren Verlauf des Meetings definiert das Entwicklungsteam operative Schritte zur Umsetzung der Anforderungen

SCRUM EVENTS: DAILY SCRUM



SCRUM EVENTS DAILY SCRUM



SCRUM EVENTS

DAILY SCRUM



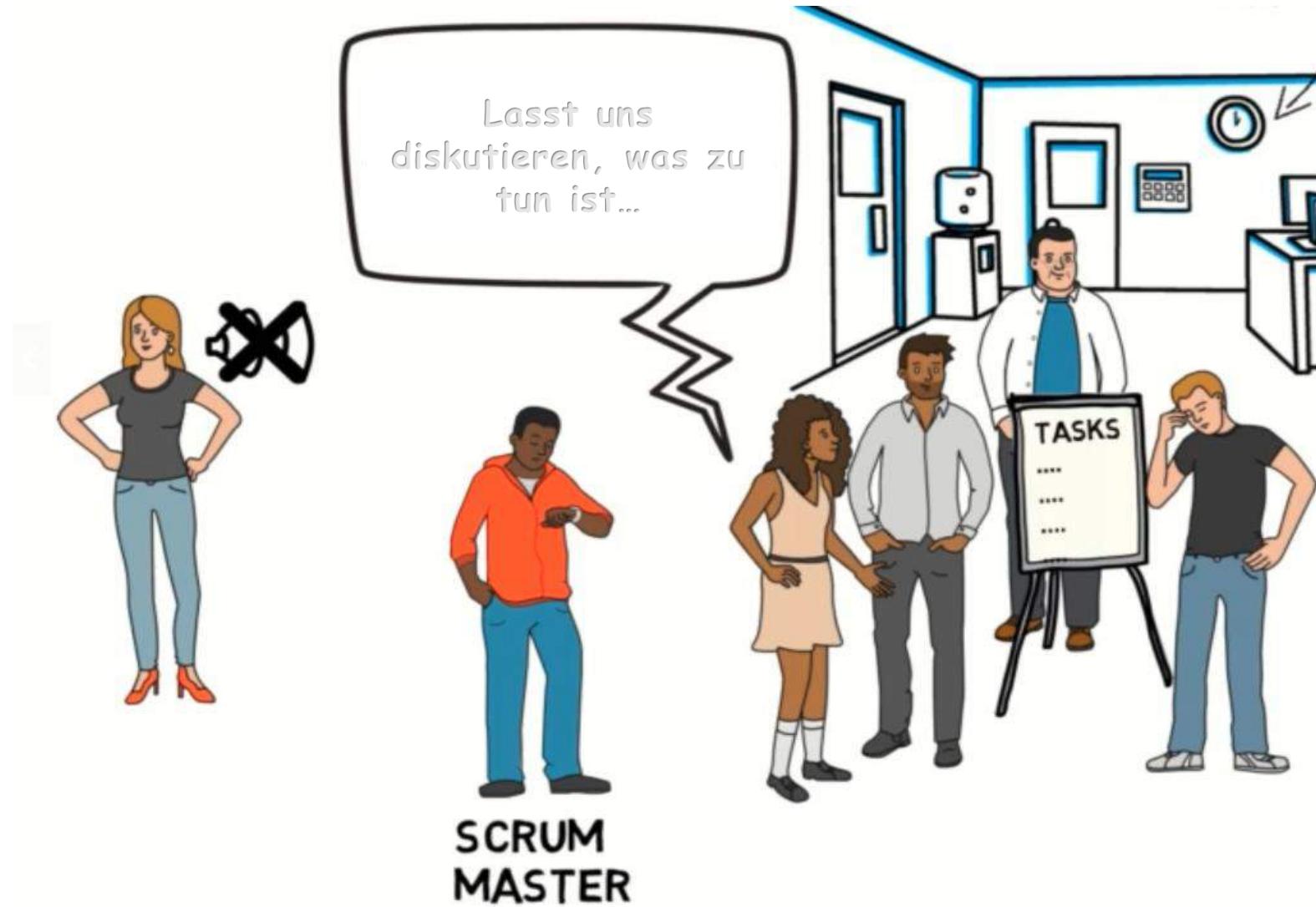
SCRUM EVENTS

DAILY SCRUM



SCRUM EVENTS

DAILY SCRUM



ZUSAMMENFASSUNG SCRUM EVENTS

DAILY SCRUM

Daily Scrum

Daily Scrum Meeting

- 15 Minuten täglich
- Entwicklungsteam
- Transparenz über derzeitige Tätigkeiten und Hindernisse

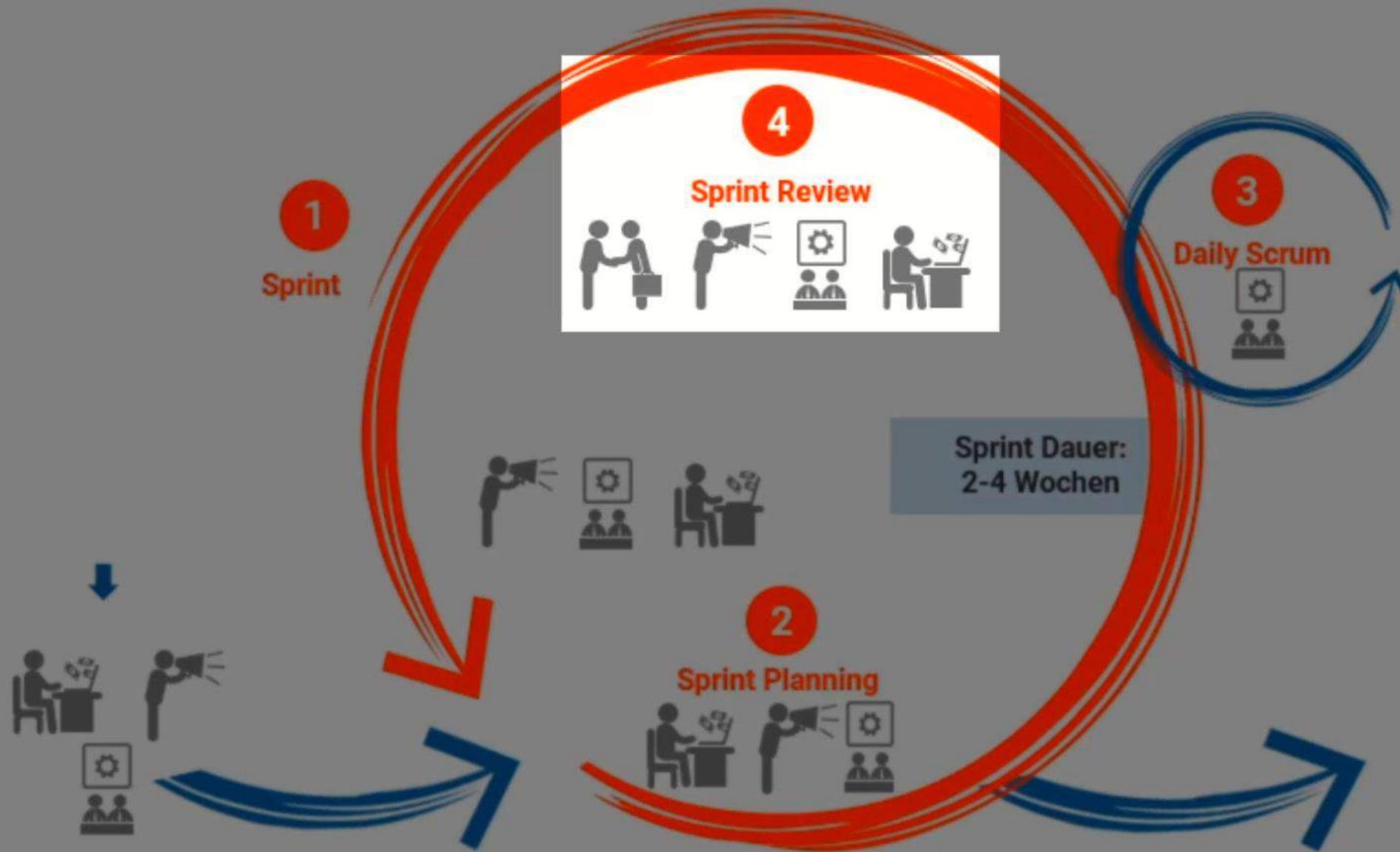
Das Daily Scrum Meeting sollte 15min nicht überschreiten und keine inhaltlichen Diskussionen enthalten. Gestaltung obliegt dem Entwicklungsteam.

Drei Fragen sind typisch:

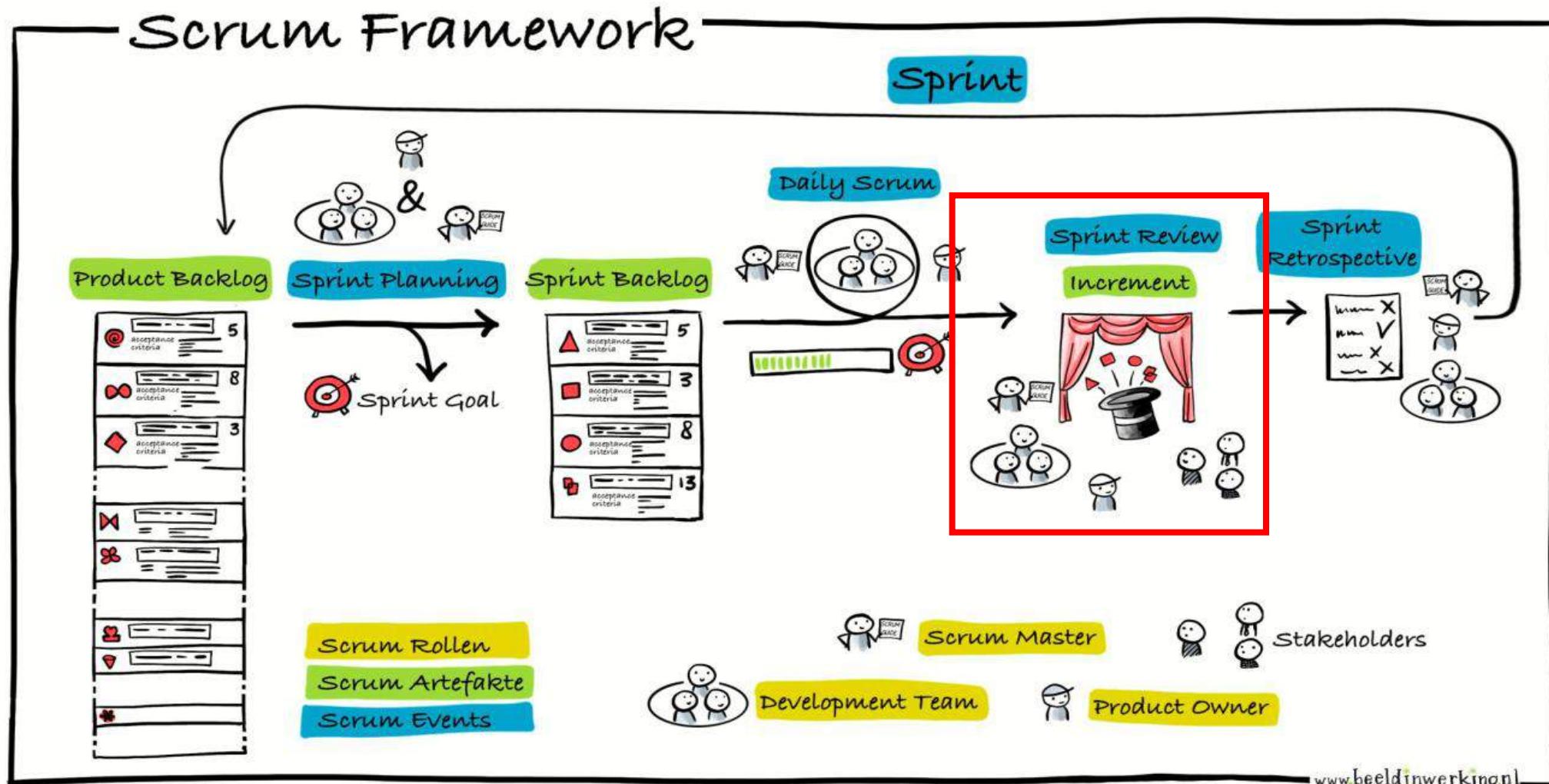
- Was habe ich seit dem letzten Daily Scrum getan?
- Was werde ich bis zum nächsten Daily Scrum tun?
- Welche Probleme und Hindernisse halten mich derzeit auf?

3

SCRUM EVENTS: SPRINT REVIEW

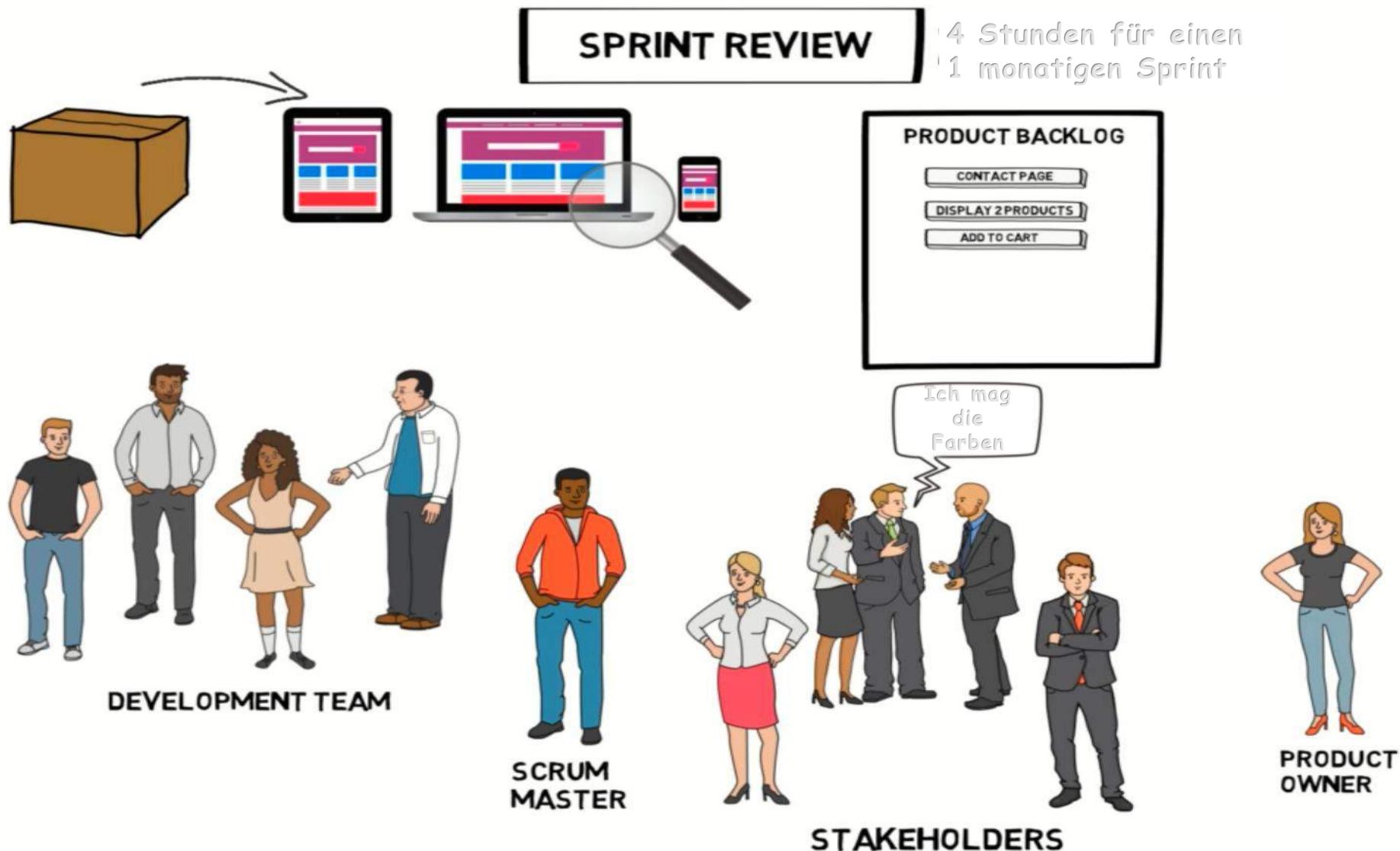


SCRUM EVENTS SPRINT REVIEW



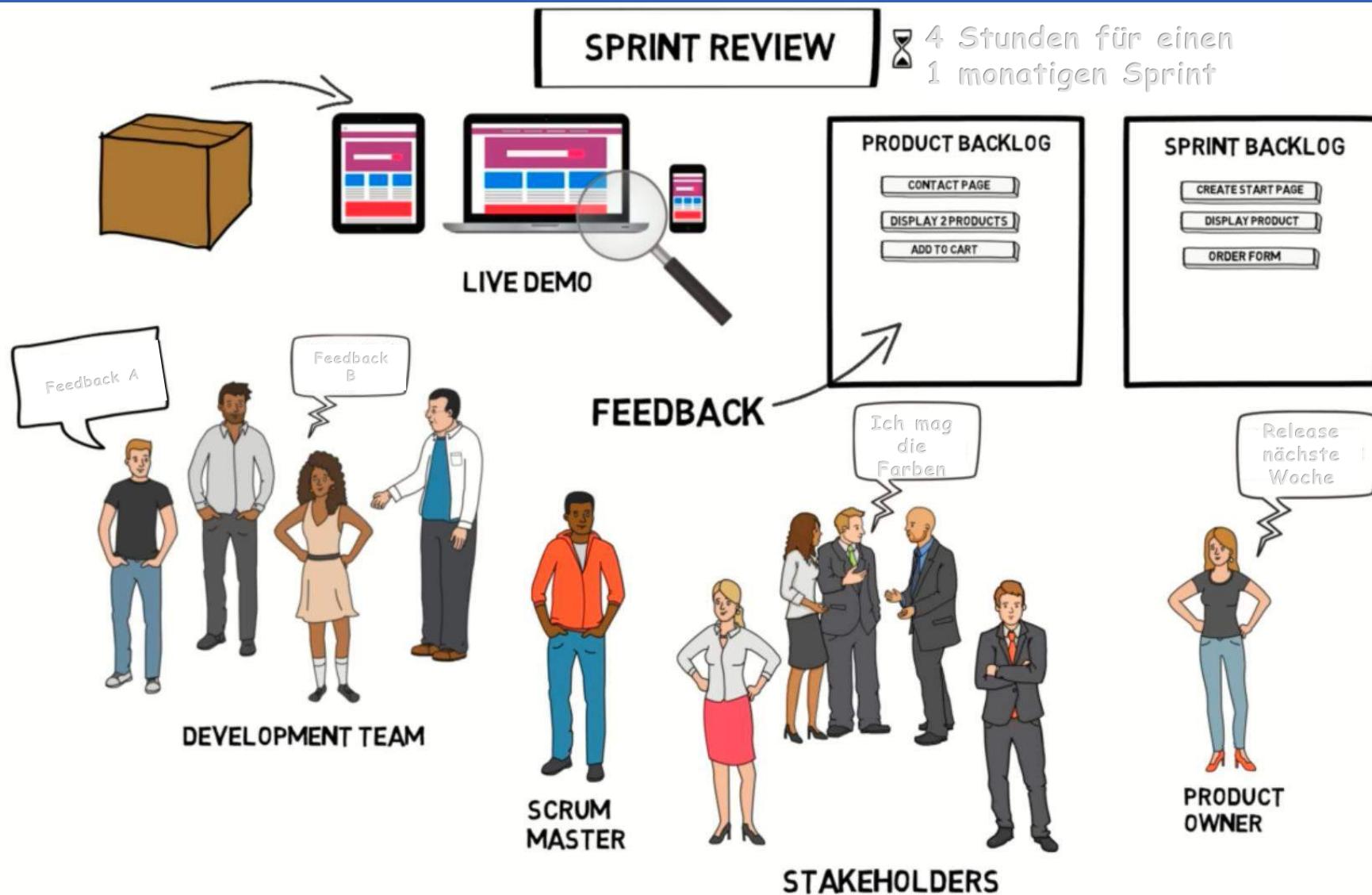
SCRUM EVENTS

SPRINT REVIEW



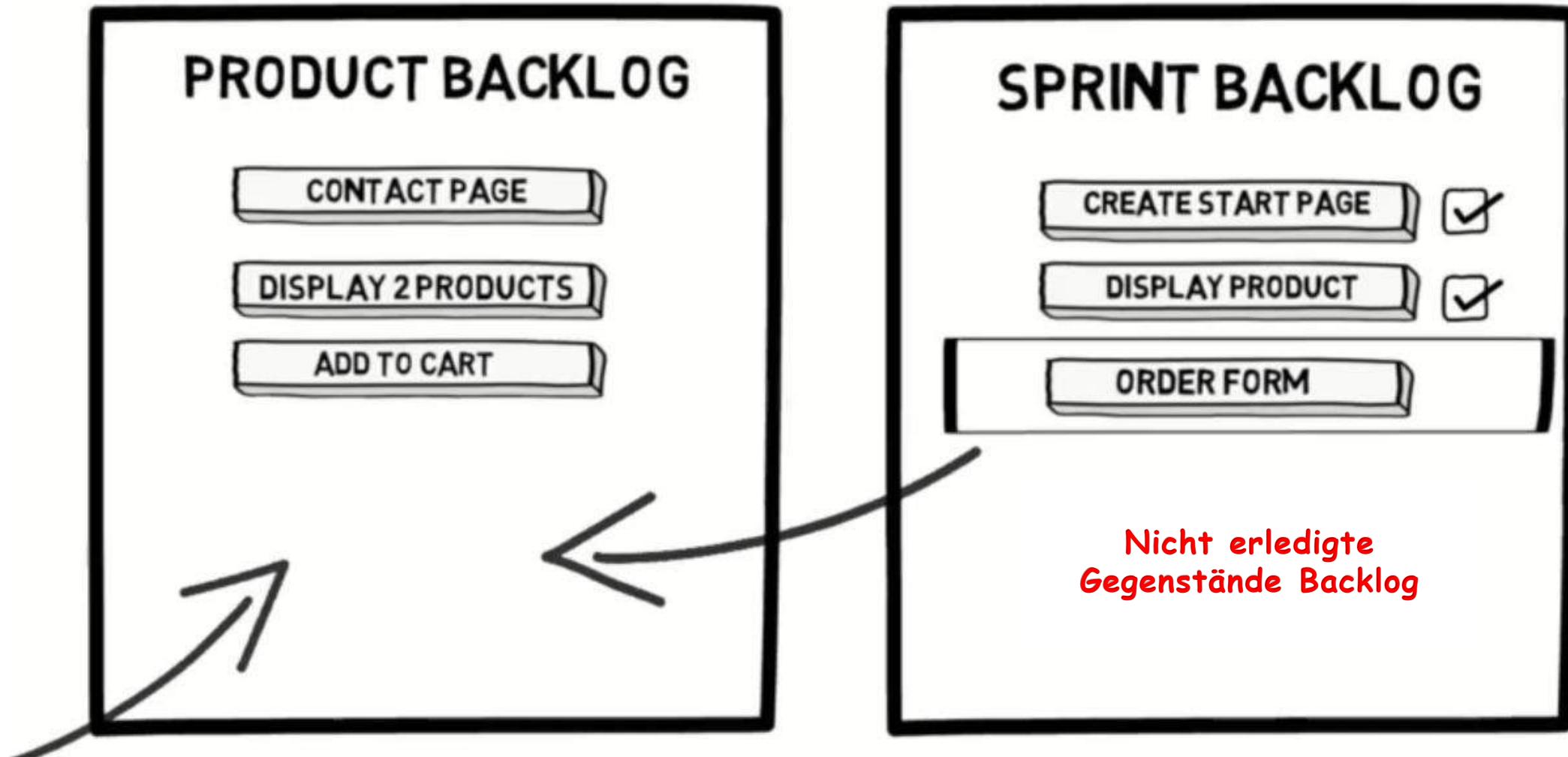
SCRUM EVENTS

SPRINT REVIEW



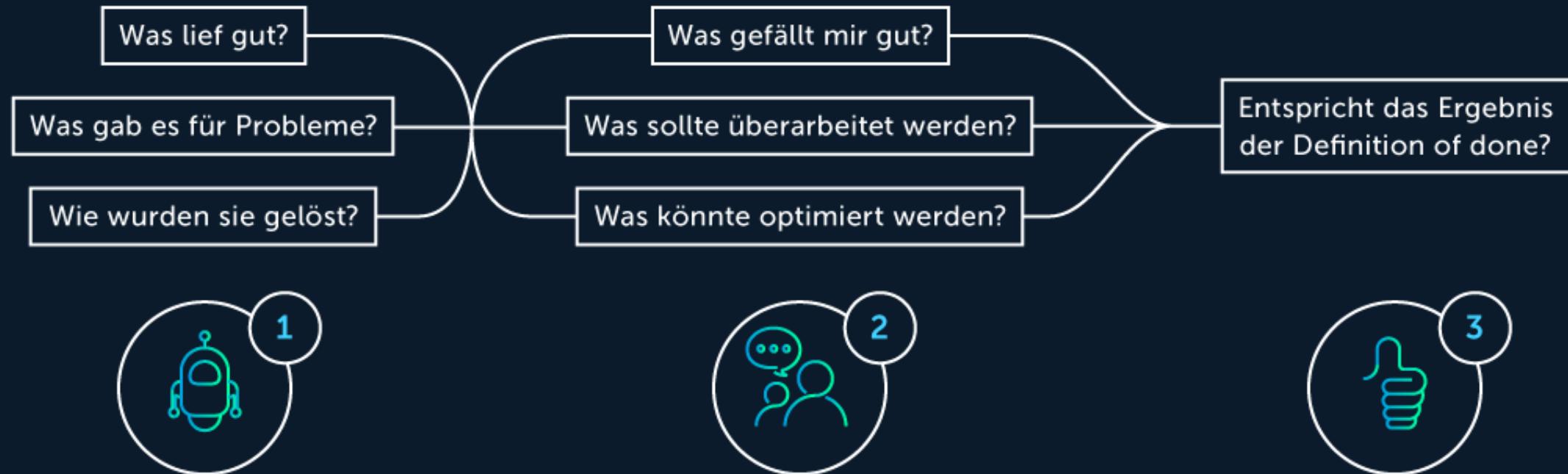
SCRUM EVENTS

SPRINT REVIEW



SCRUM EVENTS: DAS SPRINT REVIEW

Sprint Review: Ablauf der Demo



Präsentation

Das Team-Mitglied zeigt seine Ergebnisse und teilt seine Erkenntnisse aus der Entwicklung

Bewertung

Das Team, der Product Owner und alle anwesenden Stakeholder geben ihr Feedback

Freigabe

Der Product Owner entscheidet, ob das Ergebnis freigegeben wird

WARUM REVIEWS MIT KUNDEN SO WICHTIG SIND

Zweck eines Reviews ist es, bei den Projektanforderungen, die beim agilen Projektmanagement als veränderbar begriffen werden, nachzusteuern. Das oberste Ziel dabei ist eine möglichst hohe Zufriedenheit der Kunden mit dem Produkt.

Beispiel:



Für die Marketingkampagne eines Pharmaunternehmens wurden in der ersten Iteration Medikamentenaufsteller für Apotheken entwickelt. In einem Review bewerten mehrere Referenzapotheke, ob der Aufsteller für ihre Zwecke brauchbar ist. Es zeigt sich, dass viele Apotheker den Aufsteller auf ihrer Theke platzieren möchten, wozu er jedoch zu groß ist.

SCRUM EVENTS: DAS SPRINT REVIEW

7 Prinzipien der Sprint Review



Wertvolles Feedback einholen

Qualitatives Feedback im Team sorgt für eine gemeinsame Vorstellung von den zu erfüllenden Kundenbedürfnissen.



Hindernisse beseitigen

Das Team arbeitet daran Verschwendungen frühzeitig zu erkennen und zu beseitigen, damit Engagement nicht ungenutzt verpufft



Kundenmehrwert steigern

Im Fokus steht der Mehrwert für den Kunden und die Qualität der Ergebnisse mit konstruktiven Anmerkungen zu steigern.



Transparenz steigern

Die Sichtbarkeit des aktuellen Status und neuer Erkenntnisse helfen bei der Verteilung von Wissen im Unternehmen



Zusammenarbeit fördern

Mit Hilfe von kollaborativen Argumentationen und Praktiken wird Expertenwissen ausgetauscht, gleichzeitig Silodenken abgebaut.



Agile Kultur festigen

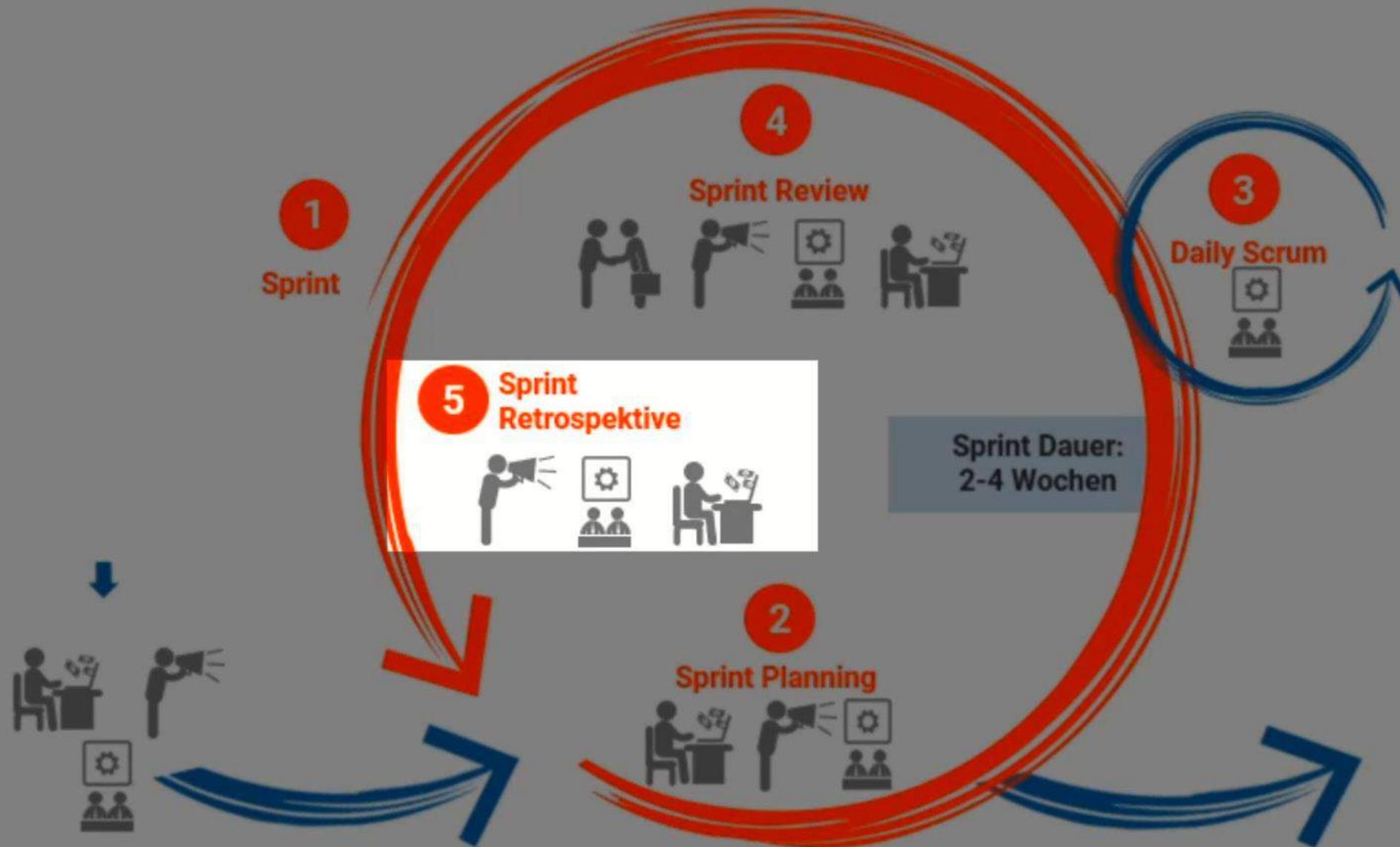
Gemeinsam nach Lösungen zu suchen sowie aus Erkenntnissen zu lernen, hilft die Fehler- und Lernkultur im Unternehmen zu verbessern.



Erfolge sichtbar machen

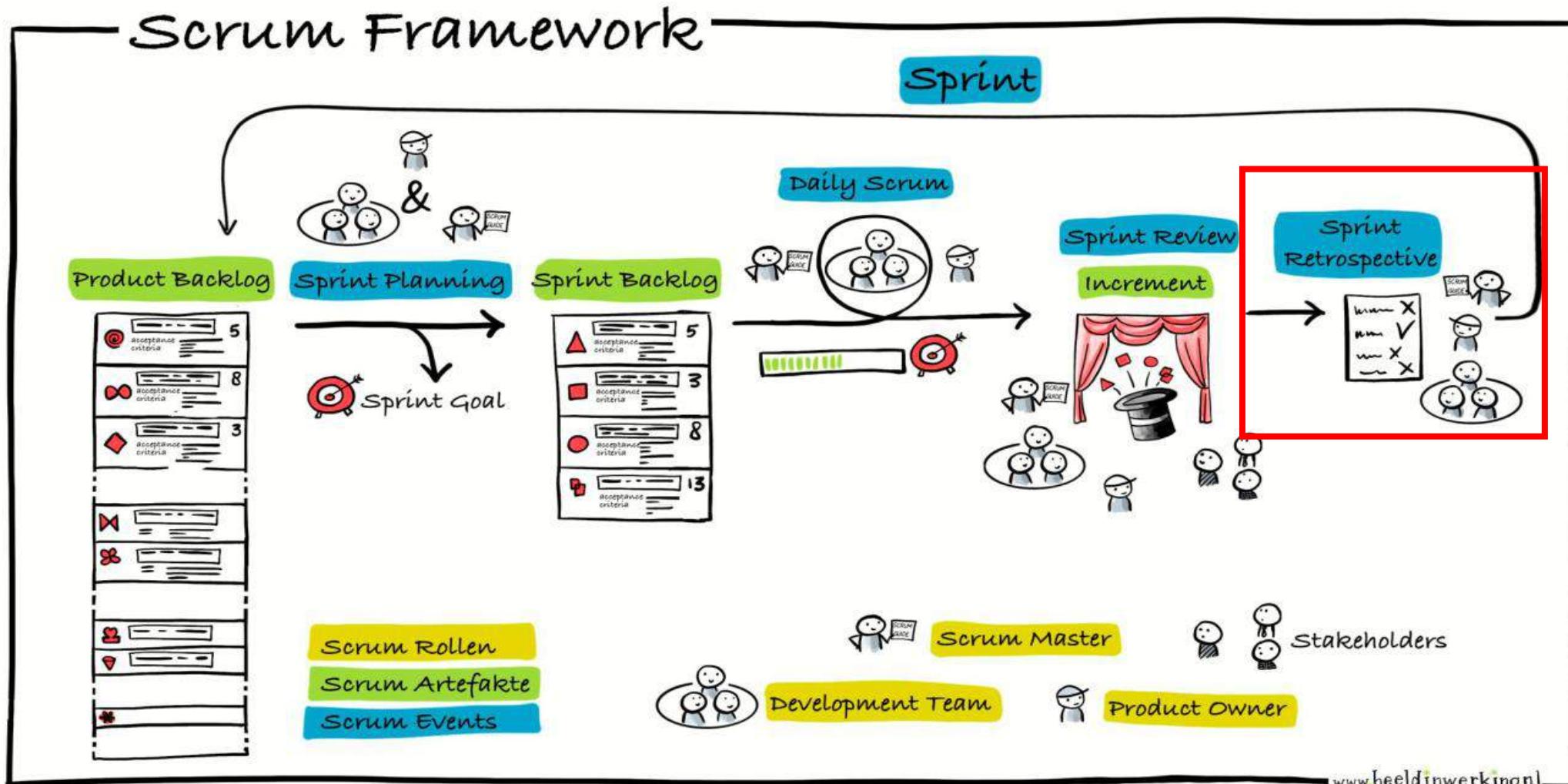
Führungskräfte und Repräsentanten können für sie relevante Informationen abholen, während das Team sich „Management Attention“ sichern kann

SCRUM EVENTS: SPRINT RETROSPEKTIVE



SCRUM EVENTS

SPRINT RETROSPECTIVE



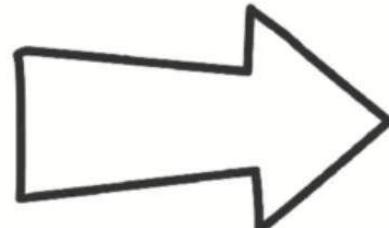
SCRUM EVENTS

SPRINT RETROSPECTIVE

SPRINT RETROSPECTIVE



INSPEKTION



PROZESS



ADAPTION

SCRUM EVENTS

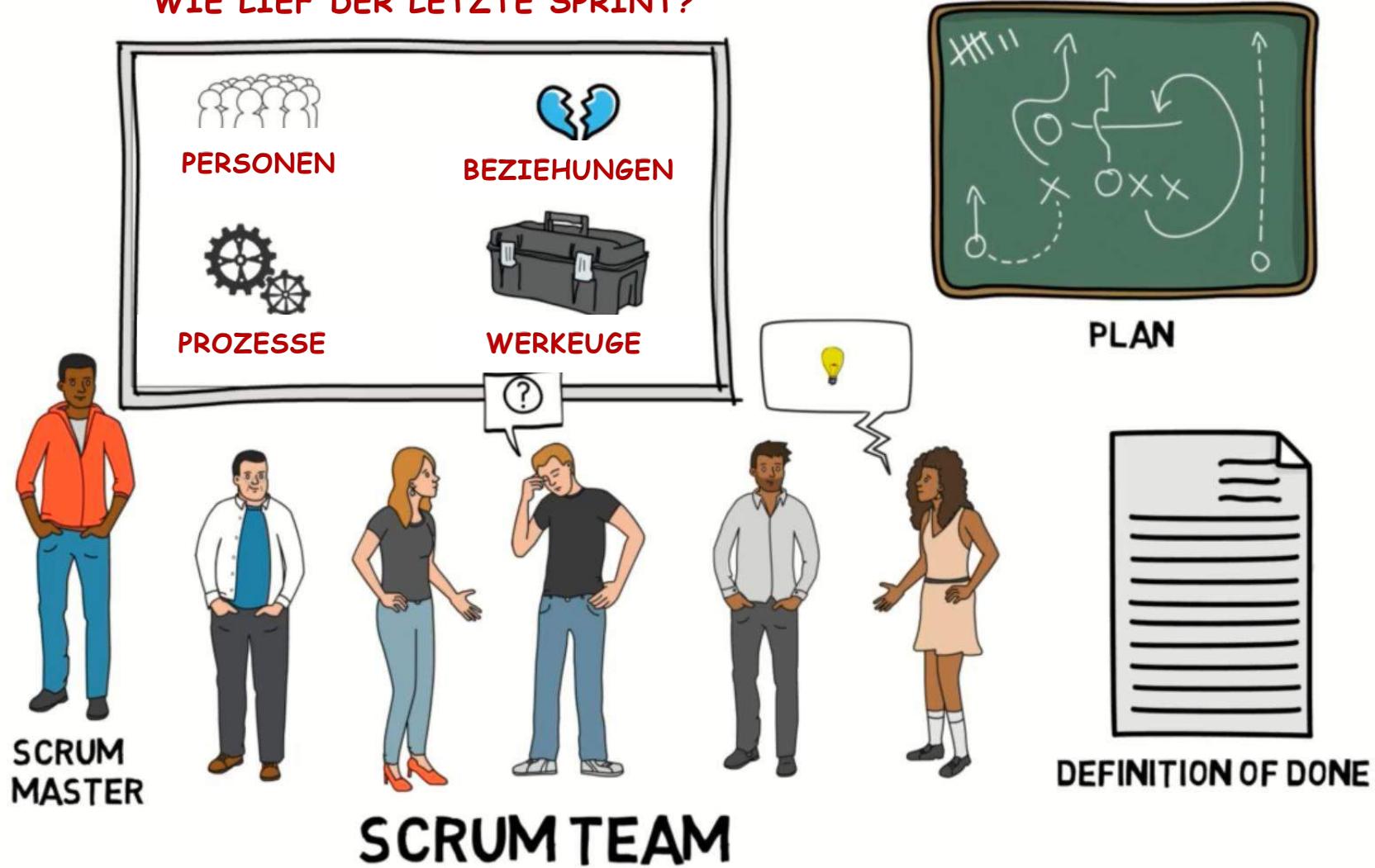
DIE FÜNF PHASEN DER SPRINT RETROSPECTIVE



SCRUM EVENTS

SPRINT RETROSPECTIVE

WIE LIEF DER LETZTE SPRINT?



ZUSAMMENFASSUNG SCRUM EVENTS

SPRINT RETROSPEKTIVE

Sprint Retrospektive

5

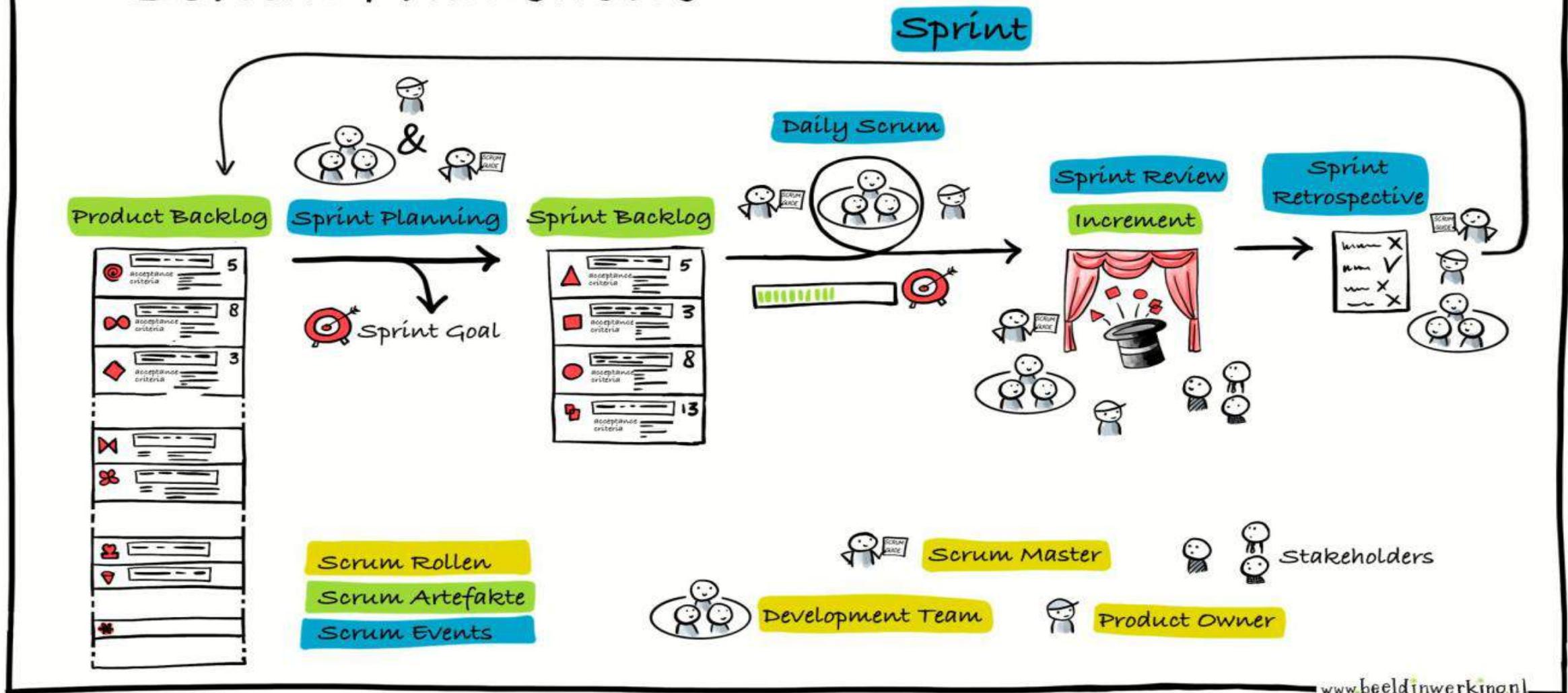
Sprint Retrospektive

- max. **3 (!)** Std. für einen einmonatig. Sprint
- Scrum Master, Entwicklungsteam, Product Owner (gem. Scrum Guide)
- Reflektiert den vergangenen Sprint (Was lief gut, was lief schlecht?) und leitet Ideen für die Verbesserung der Zusammenarbeit ab.

Die Sprint Retrospektive schließt den Sprint ab und findet unter Ausschluss der Öffentlichkeit statt. Das Team zieht sich in einen vertrauten Rahmen zurück und reflektiert den vergangenen Sprint. Offene und ehrliche Auseinandersetzungen sind der Schlüssel für die Identifikation und Behebung bestehender Schwierigkeiten

SCRUM ARTEFAKTE

Scrum Framework



SCRUM ARTEFAKTE

PRODUCT BACKLOG

Scrum besteht aus:

- 5 Aktivitäten (Ereignisse)
- 3 Rollen
- 3 Artefakte

i

1



Product
Backlog

0

Product Backlog
Refinement

1 Sprint

4

Sprint Review



5

Sprint
Retrospektive



2

Sprint Planning



3

3

Daily Scrum



Sprint Dauer:
2-4 Wochen

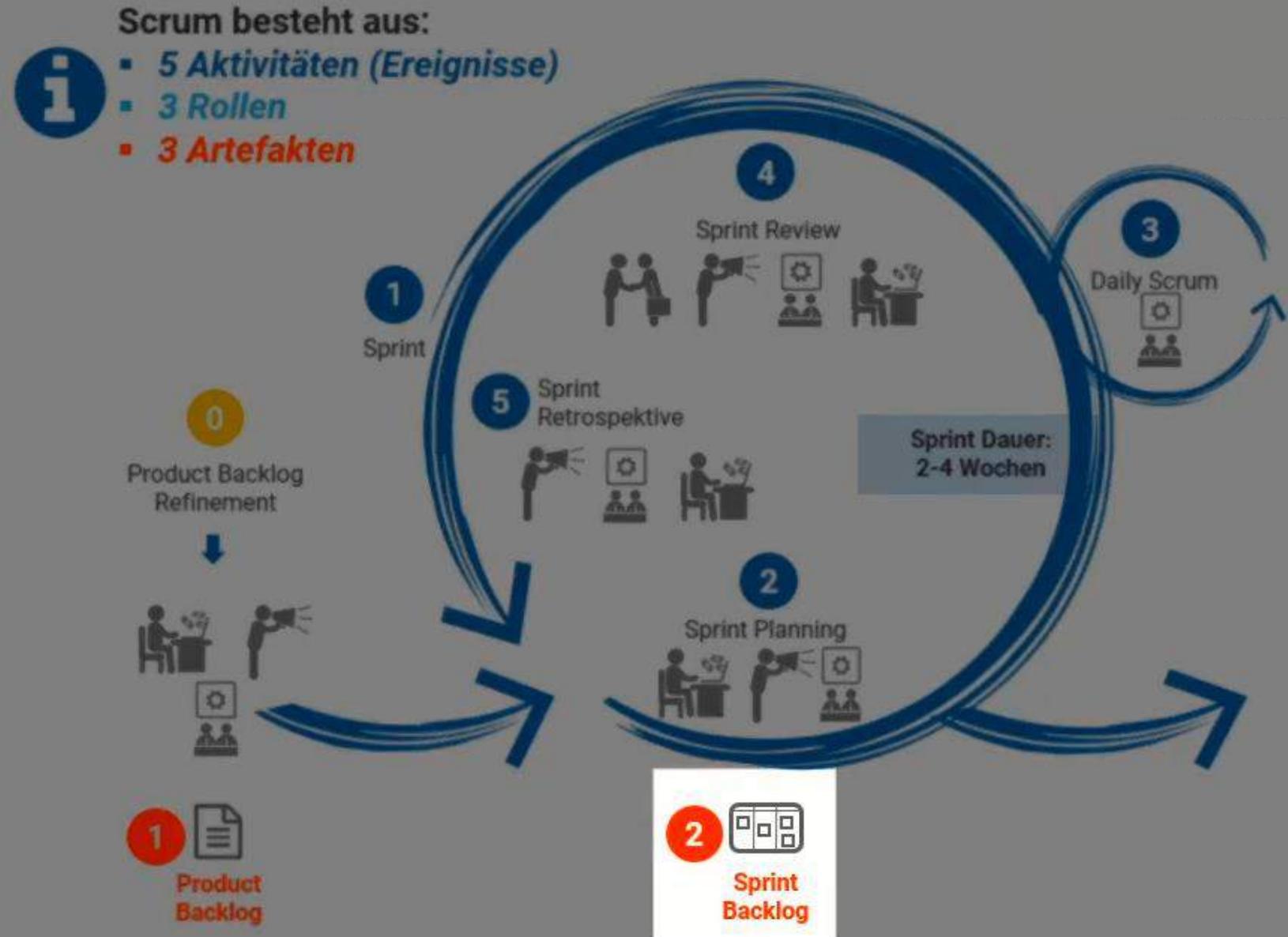


4



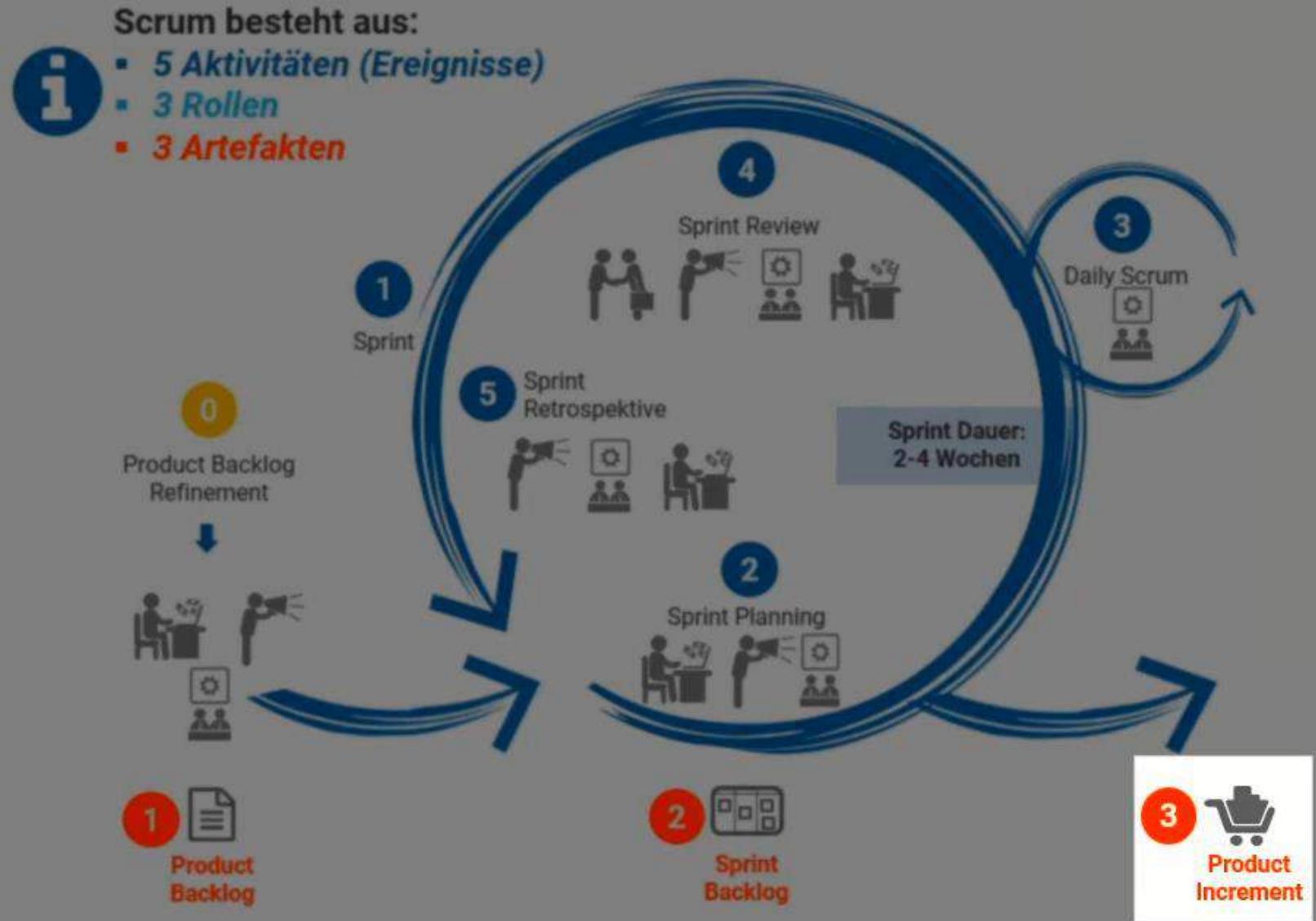
SCRUM ARTEFAKTE

SPRINT BACKLOG



SCRUM ARTEFAKTE

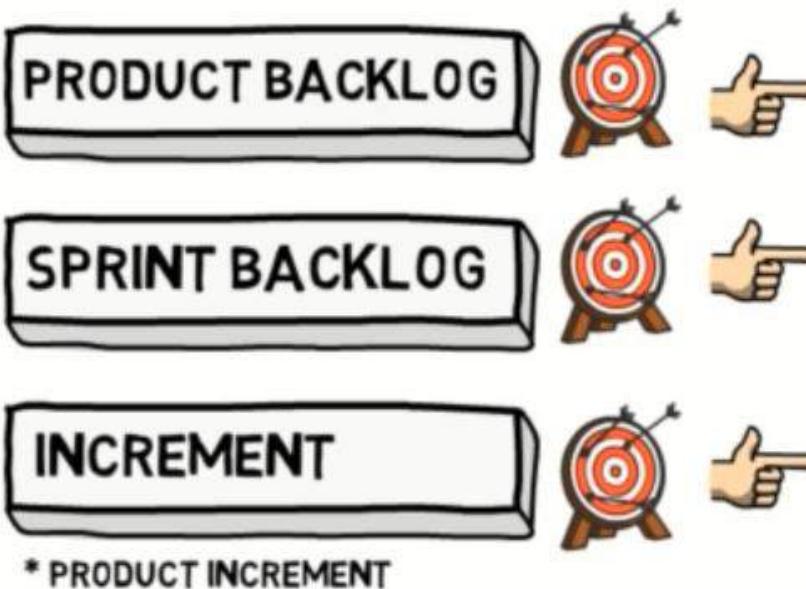
SPRINT BACKLOG



SCRUM ARTEFAKTE

- Inkrement: Es entspricht einem Teilprodukt der Entwicklung, wie es durch eine Iteration (Scrum: Sprint) entsteht.
- Product Backlog: Damit werden die für das Produkt insgesamt umzusetzenden Aufgaben bzw. Anforderungen erfasst.
- Sprint Backlog: Das sind die für die nächste Iteration (Sprint) umzusetzenden Aufgaben bzw. Anforderungen.

SCRUM ARTEFAKTE



Commitments

PRODUCT GOAL

SPRINT GOAL

DEFINITION OF DONE

👉 **WHY**
👉 **WHAT**



Goal
-> Ziel

SCRUM ARTEFAKTE & COMMITMENTS

PRODUCT GOAL
SPRINT GOAL
DEFINTION OF DONE

Commitments helfen dabei,
sicherzustellen, dass
Informationen bereitgestellt
werden, welche **Transparenz**
und **Fokus** verbessern, um
Fortschritt messbar zu machen

Commitments (2020)

Commitments helfen dabei, sicherzustellen, dass Informationen bereitgestellt werden, welche Transparenz und Fokus verbessern, um Fortschritt messbar zu machen

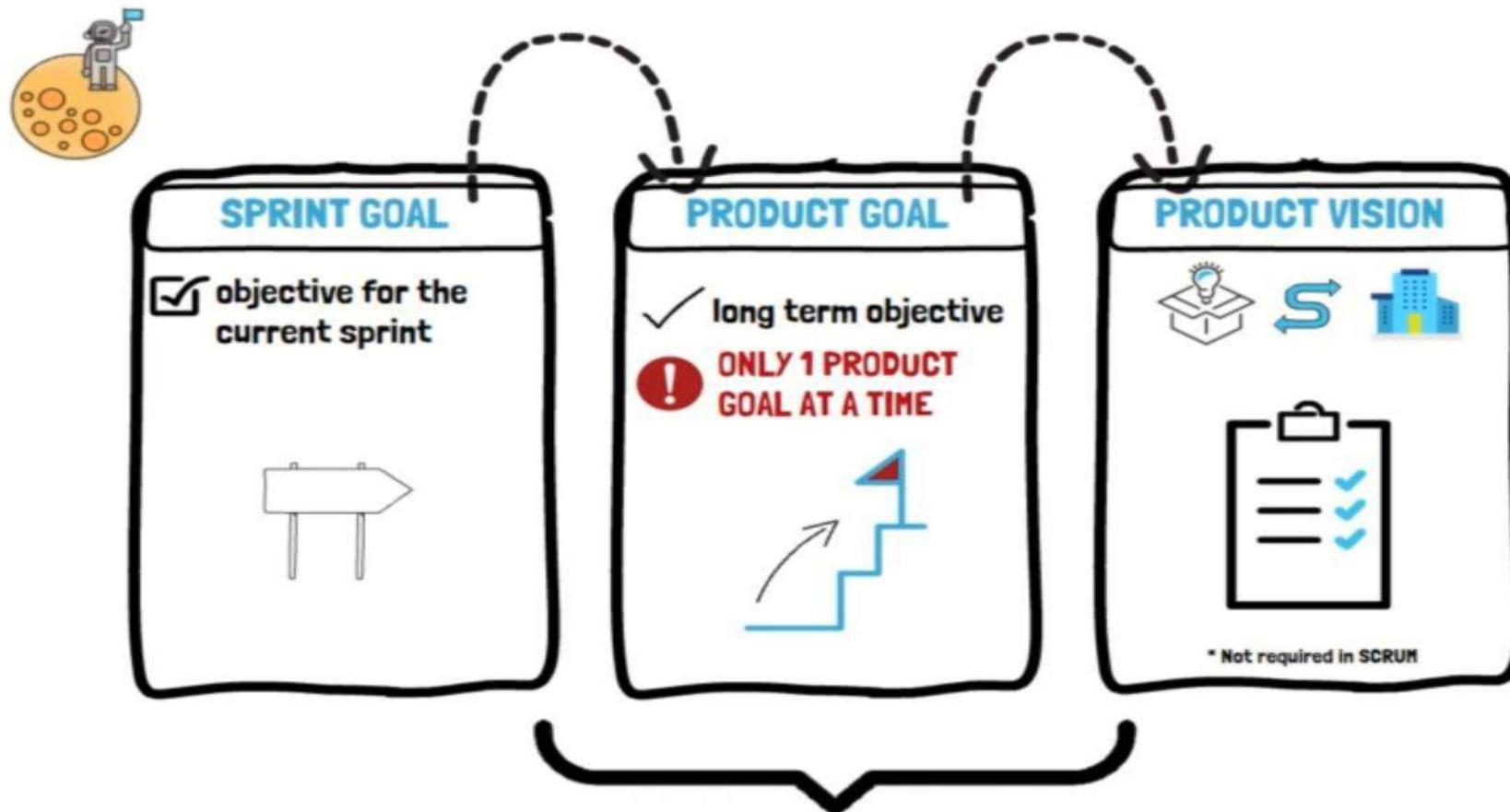
SCRUM ARTEFAKTE & COMMITMENTS

PRODUCT GOAL
SPRINT GOAL
DEFINTION OF DONE



Diese Commitments dienen dazu, Empirie und die Scrum-Werte für das Scrum Team und seine Stakeholder zu verstärken.

ZIELE UND VISIONEN



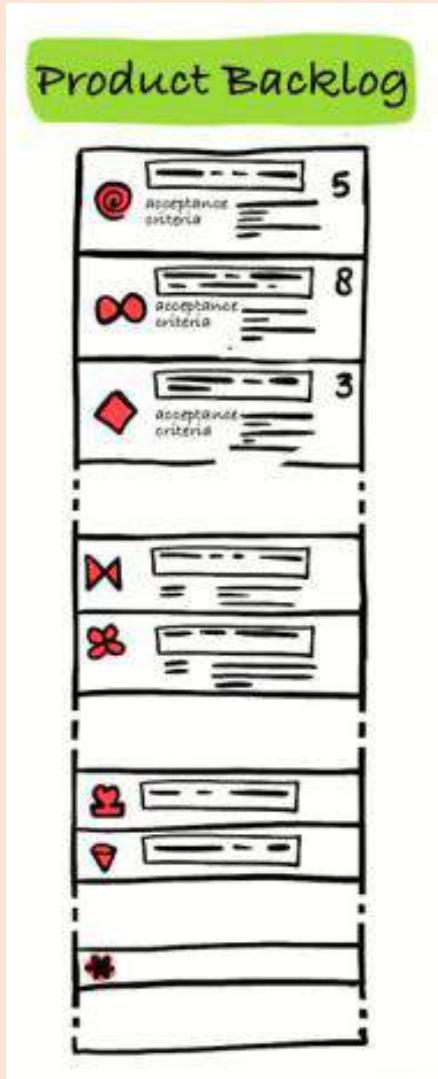
→ Transparency

→ Inspection

→ Adaptation

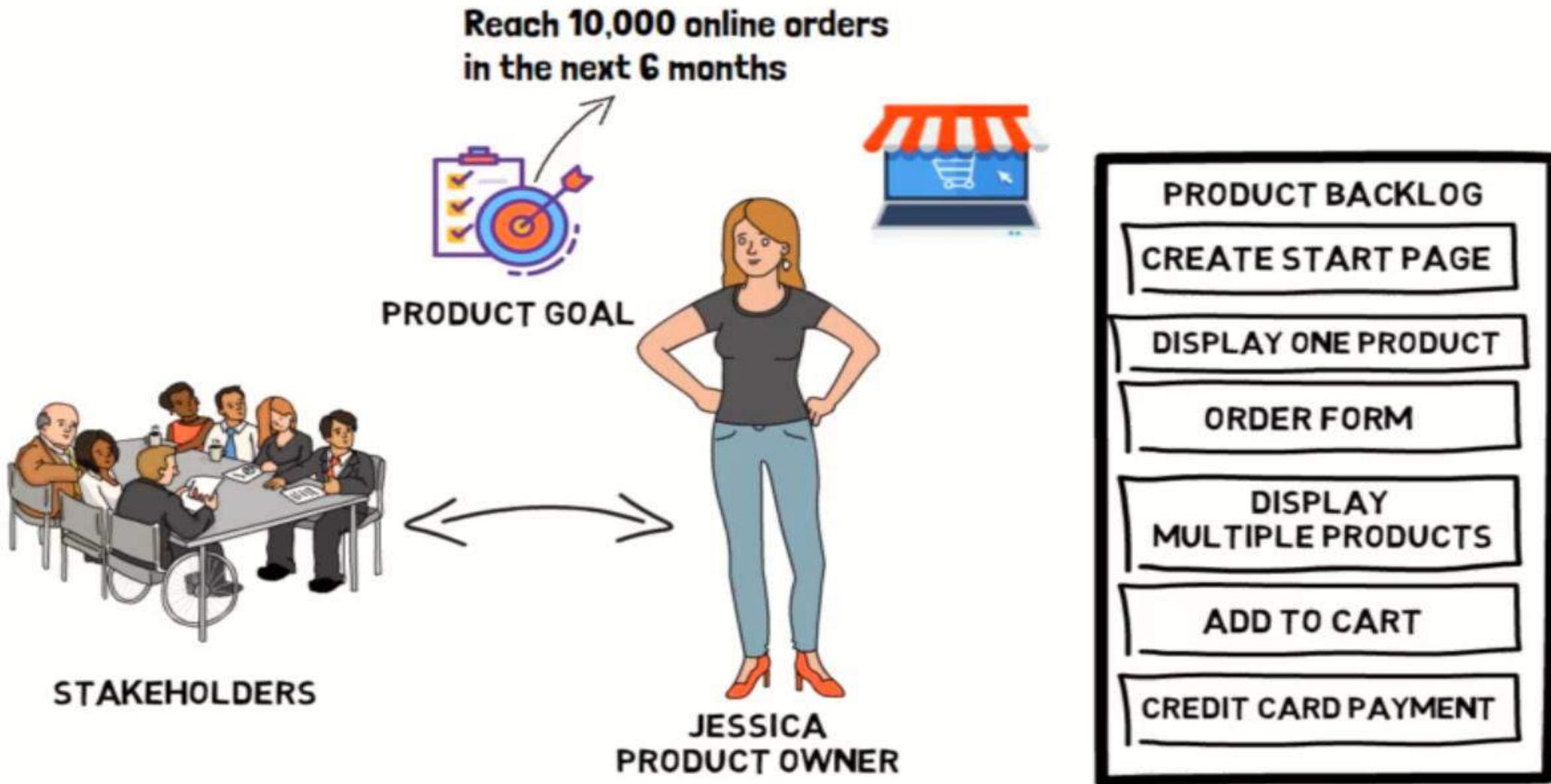
SCRUM ARTEFAKTE

PRODUCT BACKLOG



Product Owner

SCRUM ARTEFAKTE PRODUCT BACKLOG



DIE GRÖSSE VON VORHABEN

Anforderungen in Scrum

Anforderungen werden in Scrum in Form von User Stories beschrieben, ist der Reifegrad hierfür nicht ausreichend, oder zusätzliche Gruppierungen notwendig, kommen weitere Kategorien zum Einsatz



Theme

- ✓ Gruppe von Epics, die ein bestimmtes Ziel verfolgen. Kann mitunter auch ein einzelnes Produkt beschreiben

Epic

- ✓ Grobe und vorläufige Beschreibung einer Funktionalität, die im weiteren Planungsverlauf in einzelne User Stories heruntergebrochen werden muss

User Story

- ✓ Konkrete Beschreibung einer Anforderung einer gewünschten Funktionalität innerhalb des Produkts. Eine User Story enthält eine Mehrzahl von Tasks

Task

- ✓ Operative Beschreibung einzelner Aktivitäten zur Erfüllung einer User Story (bspw. einzelne Programmierschritte, erforderliche Meetings)

Nur im Sprint Backlog!

ANORDNUNG VON EPICS UND USER STORIES IM PRODUCT BACKLOG



JE NÄHER ICH AN EPIC
„OREGON“
HERANKOMME DESTO
GENAUER MUSS ICH
ÜBER DIE ETAPPEN IN
OREGON BESCHEID
WISSEN



EPICS UND USER STORIES IM SPRINT

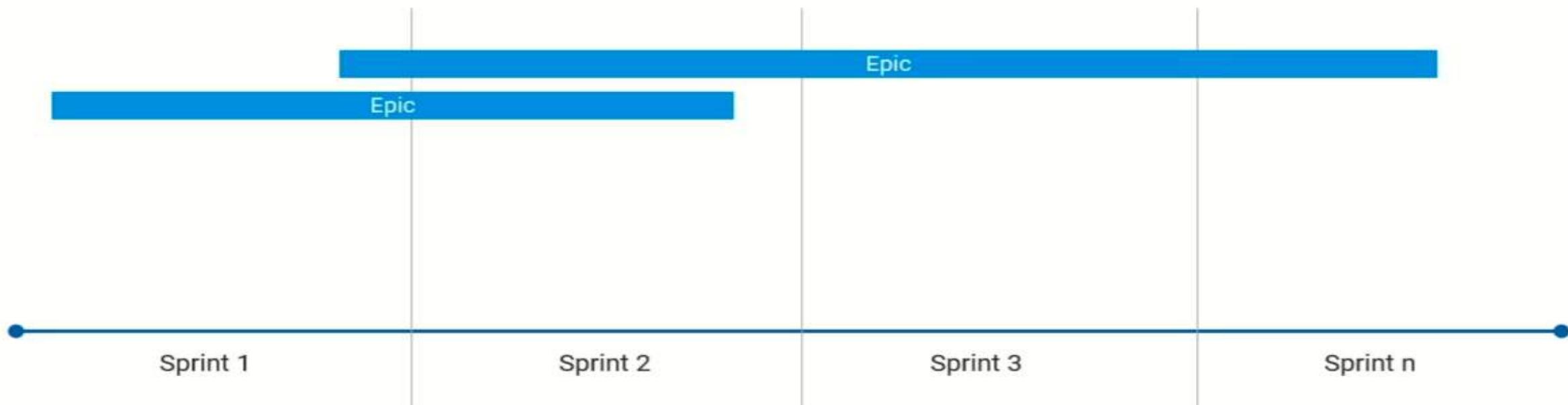
WENIGER IST MANCHMAL MEHR

Epic

Epics können mehrere Sprints umfassen und größere Unsicherheiten enthalten, sollten aber im Laufe der weiteren Planung in einzelne, klar beschriebene User Stories heruntergebrochen werden

User Story

User Stories sollten innerhalb eines Sprints umsetzbar sein um im Sprint Review ein Ergebnis demonstrieren zu können. Müssen entgegen der Epics konkret beschrieben sein und über Akzeptanzkriterien verfügen



ANORDNUNG VON EPICS UND USER STORIES IN DEN SPRINTS

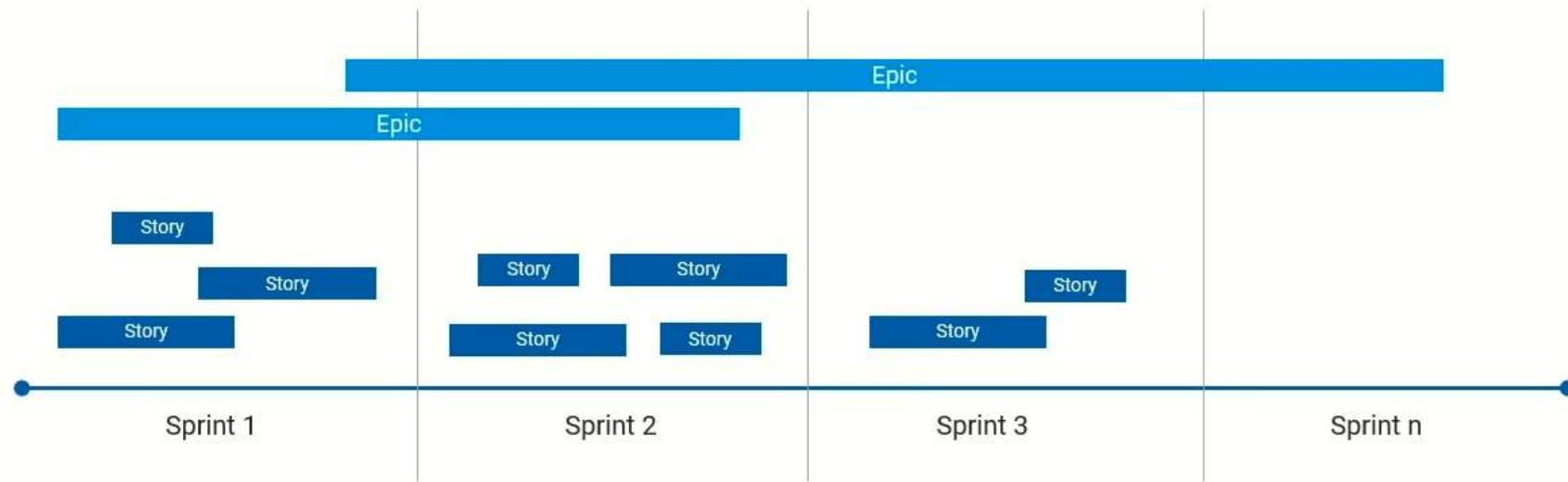
Eigenschaften von Epics und User Stories

Epic

Epics können mehrere Sprints umfassen und größere Unsicherheiten enthalten, sollten aber im Laufe der weiteren Planung in einzelne, klar beschriebene User Stories heruntergebrochen werden

User Story

User Stories sollten innerhalb eines Sprints umsetzbar sein um im Sprint Review ein Ergebnis demonstrieren zu können. Müssen entgegen der Epics konkret beschrieben sein und über Akzeptanzkriterien verfügen



DIE GRÖSSE VON VORHABEN AM BEISPIEL EINER ONLINEPRÄSENZ

Onlinepräsenz

Webshop

Blog

Events

DIE GRÖSSE VON VORHABEN AM BEISPIEL EINER ONLINEPRÄSENZ

Onlinepräsenz

Webshop

Anlage von
Benutzerkonten

Eingabemaske
bauen

Automatisierte
Bestätigungsmaile

BEISPIEL FÜR EIN OFFLINE PRODUCT / SPRINT BACKLOG

Product Backlog Beispiel

Eine visuelle Darstellung der Backlogs hilft bei Diskussionen und Priorisierungen

Erklärung

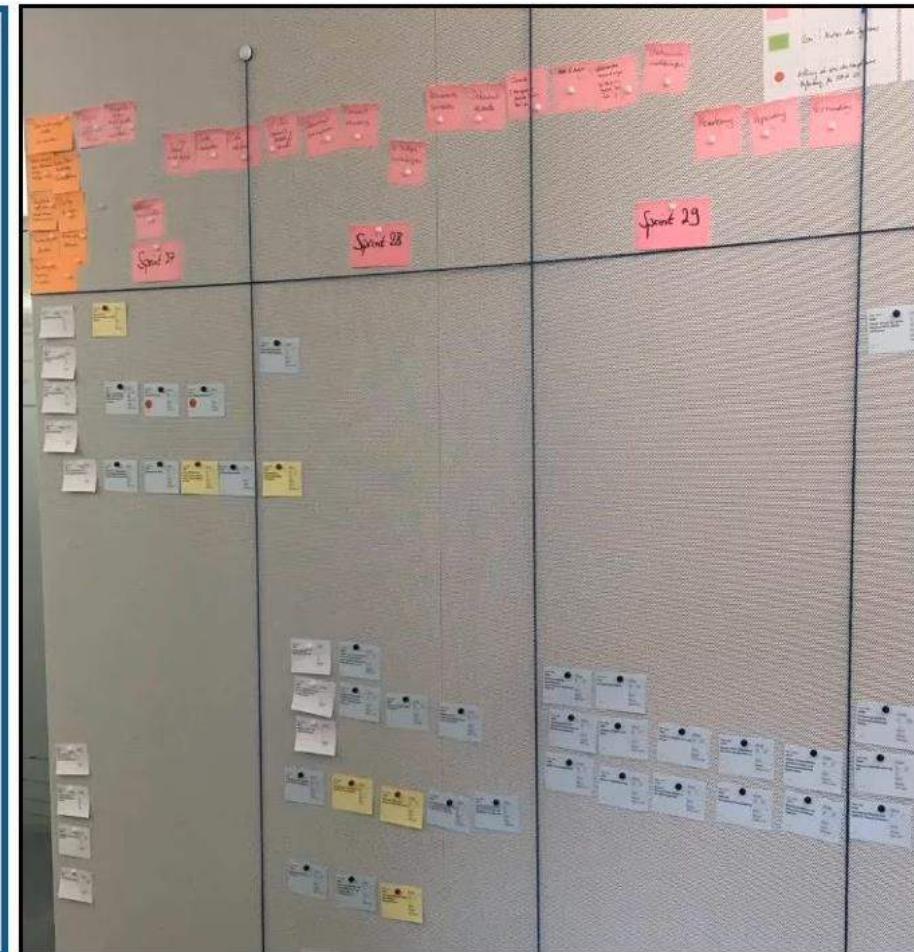
- In diesem Beispiel aus einem Projekt mit zwei Entwicklungsteams sieht man eine Aufteilung in Epics und User Stories, welche die folgenden drei Sprints abdecken

Epics

User Stories

User Stories

Issues / Bugs



EPICS UND USER STORIES

Eine User Story ist eine **einfache** und **kompakte** Beschreibung einer Anforderung.

Die Beschreibung muss erklären, welche **Funktionalität** oder **Eigenschaft** am **Ende** der Entwicklung **vorliegen** muss und für wen dies relevant ist. Zudem muss der jeweilige **Nutzen** aus der Umsetzung hervorgehen.

Erfüllen User Stories diese Kriterien noch nicht, weil Sie bspw. noch zu grob beschrieben sind, so werden diese als „**Epics**“ bezeichnet. Diese müssen dar entsprechend in einzelne konkrete User Stories heruntergebrochen werden, bevor die Umsetzung starten kann.

EPICS UND USER STORIES

Eine User Story sollte sich an folgenden Merkmalen orientieren:

Titel: <eindeutige Bezeichnung der Anforderung>

Als <eindeutige Beschreibung des Anfordernden oder der Persona>

Möchte ich <Beschreibung der Anforderung / Tätigkeit>

Damit <Beschreibung des Nutzens>

Beispiel:
User Accounts anlegen

Als Administrator möchte ich in der Lage sein User Accounts für andere Nutzer anzulegen **damit** diese später problemlos auf das System zugreifen können um damit zu arbeiten

SCRUM ARTEFAKTE

PRODUCT BACKLOG / USER STORYS

Als <Rolle>
möchte ich <Funktionalität>,
um <Nutzen>
zu erreichen.

Akzeptanzkriterien:

.....
.....

EPICS UND USER STORIES

Eine User Story **beschreibt nicht „Wie“ die Umsetzung zu erfolgen hat**. Dies liegt in der Verantwortung des **Entwicklungsteam** und wird **eigenverantwortlich** und **selbst-organisiert** vom Entwicklungsteam geplant.

User Stories sind als „**Einladung zur Diskussion**“ zu verstehen und sollen die direkte Diskussion zwischen Product Owner und dem Entwicklungsteam und etwaigen Stakeholdern fördern, bis alle Beteiligten eine klare Vorstellung von der Anforderung und den erforderlichen Umsetzungsschritten haben.

Gute User Stories folgen der INVEST-Regel

EPIC ODER USER STORY?

Gute User Stories folgen der „**INVEST-Regel**“

- „**I**“-ndependent:
möglichst unabhängig von anderen User Stories
- „**N**“-egotiable:
User Stories müssen „verhandelbar“ sein, bzw. diskutierbar sein
- „**V**“-aluable:
User Stories müssen einen Business Nutzen haben
- „**E**“-stimated:
User Stories müssen geschätzt sein
- „**S**“-ized
Angemessen groß: User Stories müssen innerhalb eines Sprints umsetzbar sein
- „**T**“-estable:
Das Ergebnis aus der User Story muss konkret getestet werden können

USER STORIES

AUF EINEN BLICK

User Stories schreiben

Eine User Story sollte sich an folgenden Merkmalen orientieren:

Titel: <einheitliche Bezeichnung der Anforderung>

Als <einheitliche Beschreibung des Anfordernden oder der Persona>

Möchte ich <Beschreibung der Anforderung / Tätigkeit>

Damit <Beschreibung des Nutzens>

Hilfestellung für gute User Stories

Beispiel:
User Accounts anlegen

Als Administrator möchte ich in der Lage sein User Accounts für andere Nutzer anzulegen damit diese später problemlos auf das System zugreifen können um damit zu arbeiten

INVEST-Regel

- Independent
- Negotiable
- Valuable
- Estimated
- Sized
- Testable

Akzeptanzkriterien

Individuelle Testfälle und/oder Eigenschaften für einzelne User Stories die nach Umsetzung der User Story erfolgreich, oder erfüllt sein müssen

User Stories sollten stets aus der Sicht des „**Anwenders**“ geschrieben werden und sich auf die **Beschreibung des Problems** und **nicht** der **Lösung** fokussieren!

EPIC ODER USER STORY?

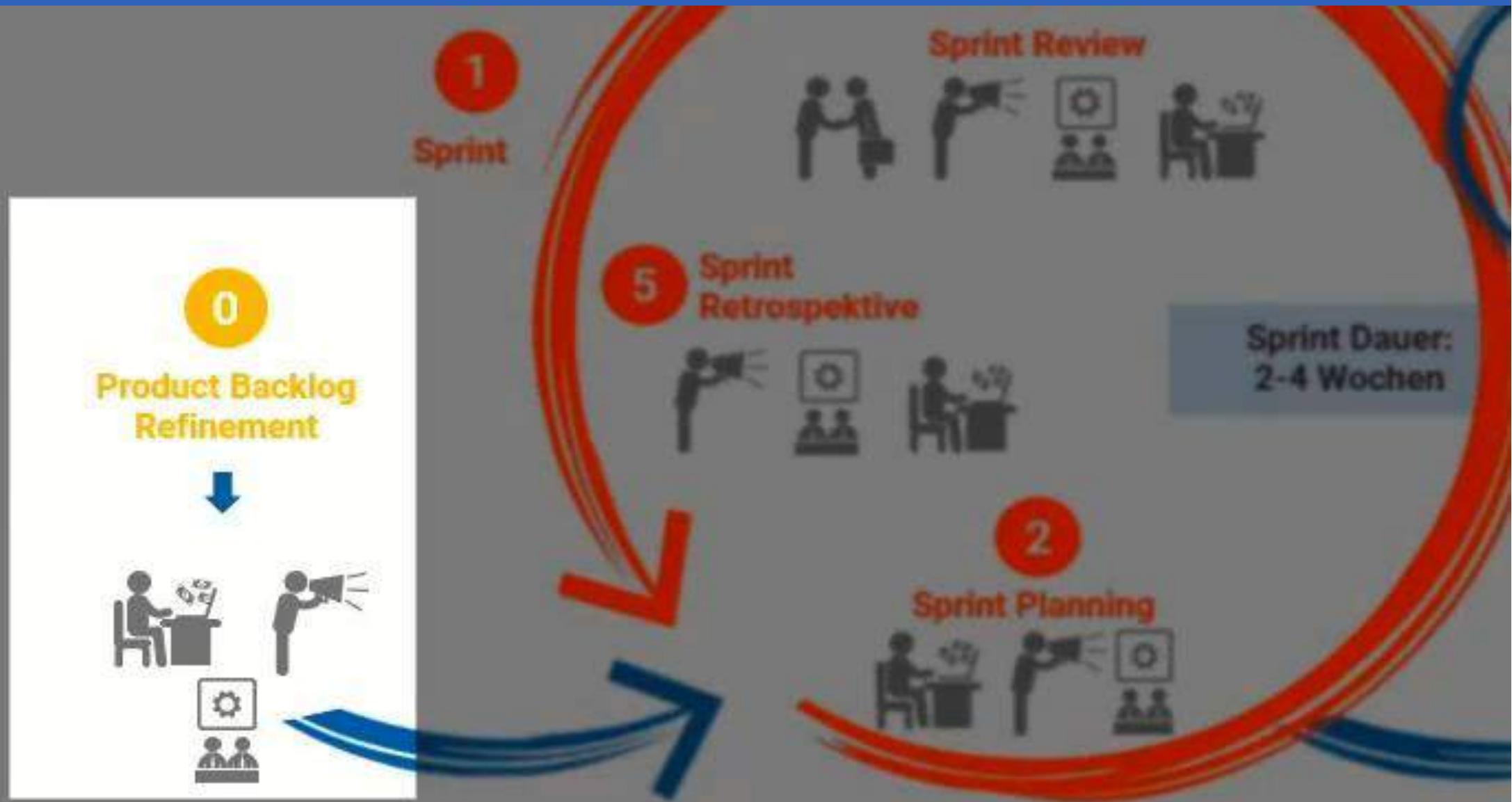
Epic

Ein Epic dient als **Platzhalter für eine konkretere Beschreibung** der Anforderung (User Story) zu einem späteren Zeitpunkt

Epics unterscheiden sich von User Stories durch:

- **Komplexität:**
Die Anforderung ist (noch) zu komplex, oder unverständlich um eine konkrete Umsetzung zu starten. Es bestehen Rückfragen
- **Unbekanntheit:**
Die Anforderung ist dem Team bislang nicht bekannt und es konnten keine Rückfragen gestellt werden
- **Risiko:**
Die Umsetzung ist mit Risiken behaftet, deren Auswirkung noch nicht abschließend geklärt wurde
- **Größe:**
Die Anforderung ist zu groß um innerhalb eines Sprints umgesetzt zu werden

KEIN OFFIZIELLES SCRUM EVENT: PRODUCT BACKLOG REFINEMENT



KEIN OFFIZIELLES SCRUM EVENT: PRODUCT BACKLOG REFINEMENT

(Product Backlog Refinement)

Das Product Backlog Refinement dient zur Überprüfung des Product Backlogs



Product
Owner



Scrum
Master



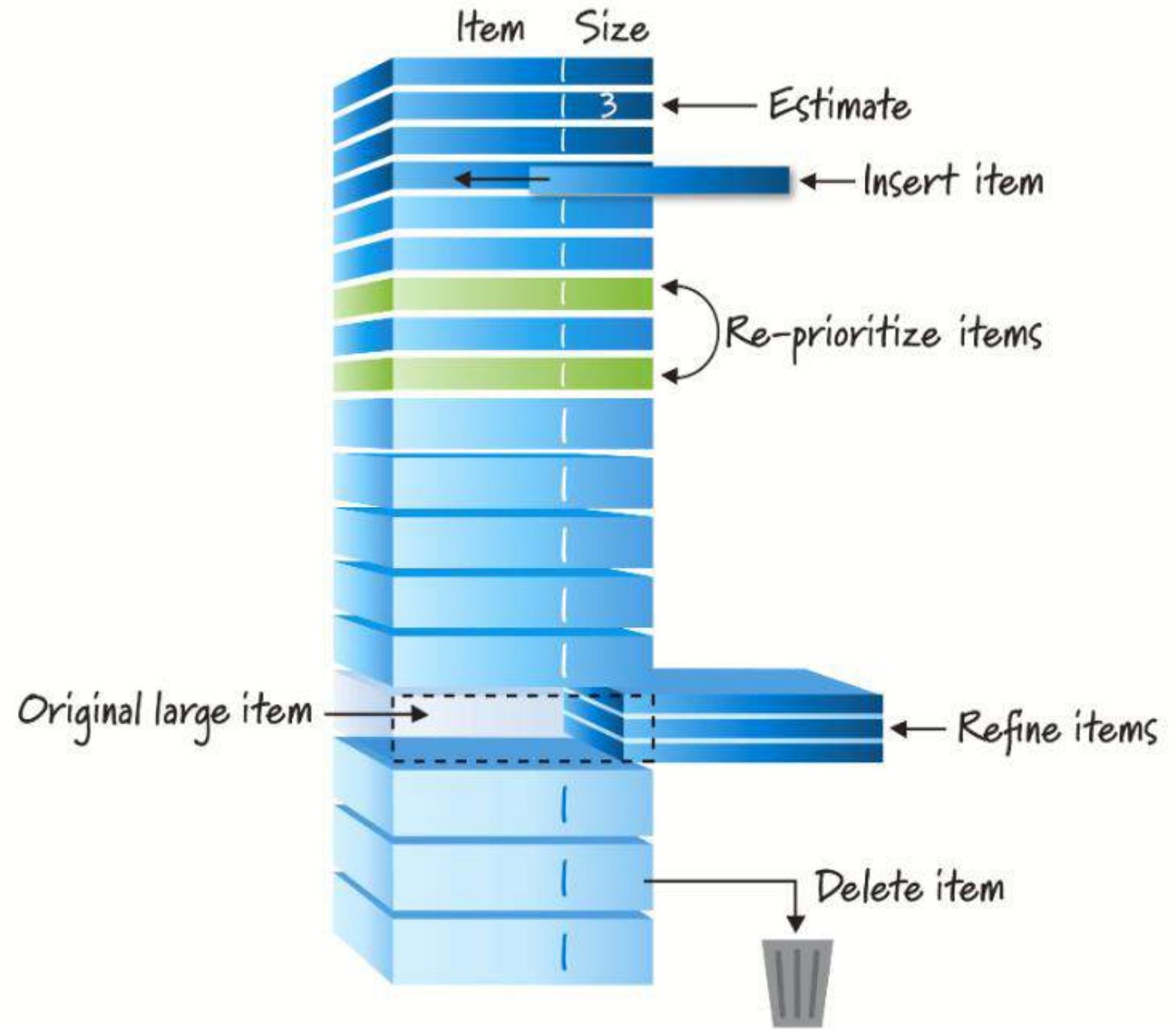
Entwicklungs-
team



- Zum Projektbeginn öfters notwendig

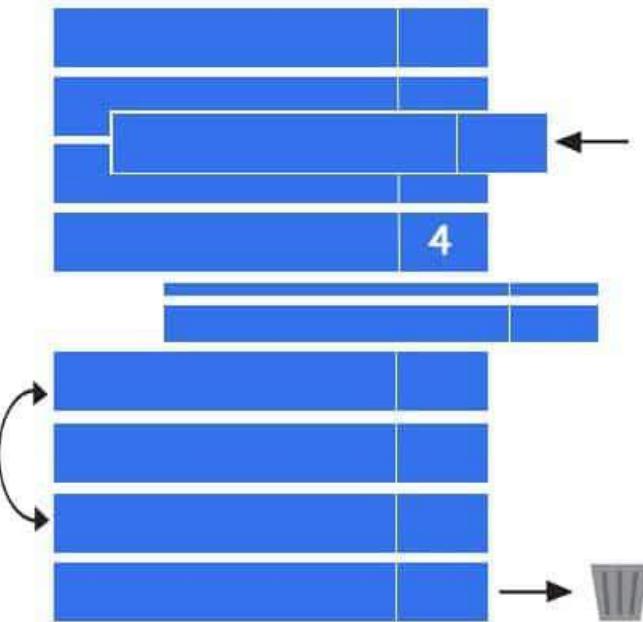
Achtung: Im Update des Scrum, Guides vom November 2020 wurde die 10% Zeitangabe endlich **ersatzlos gestrichen**

DAS BACKLOG REFINEMENT



DAS BACKLOG REFINEMENT

- Aufnahme von neuen User Storys, Features und Epics in das Backlog
- Verfeinerung von vorhandenen Epics, Features und User Storys und gegebenenfalls Aufteilung in mehrere kleine Backlog Items
- Zusammenfassen von User Storys
- Diskussion der Akzeptanzkriterien, Annahmen und Einschränkungen
- Identifikation von Abhängigkeiten
- Korrektur von Aufwandsschätzung aufgrund neuer Erkenntnisse
- Veränderung der Priorisierung aufgrund neuer Erkenntnisse
- Entfernen von irrelevanten User Storys, Features oder Epics aus dem Backlog
- Beseitigung von Unklarheiten durch gemeinsame Diskussion, wobei es sich um das „was“ und nicht das „wie“ geht



ZUSAMMENFASSUNG SCRUM EVENTS

PRODUCT BACKLOG REFINEMENT

Product Backlog Refinement

0

Product Backlog Refinement (kein offizielles Event!!)

- Scrum Team (PO, Entw.-team, Scrum Master) + Stakeholders
- Erstellt das initiale Backlog und pflegt dieses im weiteren Verlauf

Das Product Backlog Refinement sollte insb. zu Beginn des Projekts regelmäßig (1-2x pro Sprint) durchgeführt werden, um Klarheit über die bestehenden Anforderungen zu schaffen und diese ordentlich zu priorisieren, zu beschreiben und zu schätzen. Detailplanungen zur operativen Umsetzung erfolgen im Sprint Planning

DEFINITION OF READY: KÖNNEN WIR ENDLICH LOSLEGEN

Definition of Ready

Die „Definition of Ready“ bezeichnet eine Liste von Kriterien, die erfüllt sein müssen bevor eine User Story in die Umsetzung gehen kann.

Die „Definition of Ready“ ist eine einfache Liste mit Merkmalen und Kriterien, die als Qualitätssiegel für den Reifegrad einer User Story genutzt werden kann. Die Definition of Ready fragt nach jenen Kriterien um eine möglichst reibungslose Umsetzung innerhalb des Sprints zu ermöglichen und hilft dabei, wichtige Fragen im Vorfeld zu klären.



DEFINITION OF READY

KRITERIEN

Definition of Ready

- Beschreibt Kriterien die User Stories erfüllen müssen, bevor die Umsetzung beginnen kann
- Die Definition of Ready ist für alle User Stories gültig und bezieht sich nicht auf konkrete User Story Inhalte
- Funktioniert als „Checklist“ für den Product Owner und das Entwicklungsteam um wichtige Faktoren frühzeitig zu klären und keine Zeit innerhalb der Sprints zu verlieren
- Eine verbindliche „Definition of Ready“ bringt viel Struktur und Sicherheit in das Projekt und reduziert die Zeit späterer Nacharbeiten aufgrund ungeklärter Fragen
- Die „Definition of Ready“ sollte gemeinsam vom Product Owner und dem Entwicklungsteam erarbeitet werden und kann je nach Team und Unternehmen variieren. Der Scrum Master sollte bei der Ausarbeitung unterstützen und seine Erfahrungen einfließen lassen
- Die „Definition of Ready“ muss allen Beteiligten bekannt sein und sollte öffentlich ausgehängt werden

DEFINITION OF READY

BEISPIEL

Definition of Ready

- Ein Beispiel -

- **Die User Story...**
 - ist von allen Beteiligten (Stakeholdern, Entwicklungsteam, Product Owner) verstanden und alle relevanten Informationen liegen dem Entwicklungsteam vor (Zeichnungen, Screenshots, Diskussionsnotizen...)
 - Erfüllt die „INVEST“ Kriterien
 - Verfügt über festgelegte Akzeptanzkriterien, die ebenfalls von allen Beteiligten verstanden und akzeptiert sind
 - Verfügt über eine Aufwandsschätzung
 - Ist priorisiert
 - Ist innerhalb eines Sprints umsetzbar
 - Benötigt keine externe Unterstützung (Dienstleister, o. anderer Fachbereich), oder die externe Unterstützung ist für den geplanten Sprint zugesagt
 - Alle derzeit offenen Fragen wurden geklärt und einer Umsetzung stehen keine weiteren Faktoren im Weg.

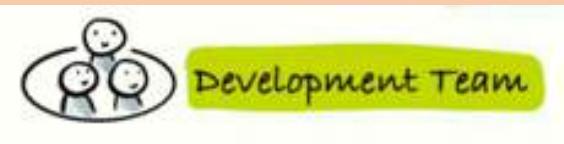
SCRUM ARTEFAKTE PRODUCT BACKLOG



WIR SIND
VERANTWORTLICH
FÜR:

- Die Einschätzung des Aufwands (Sizing)

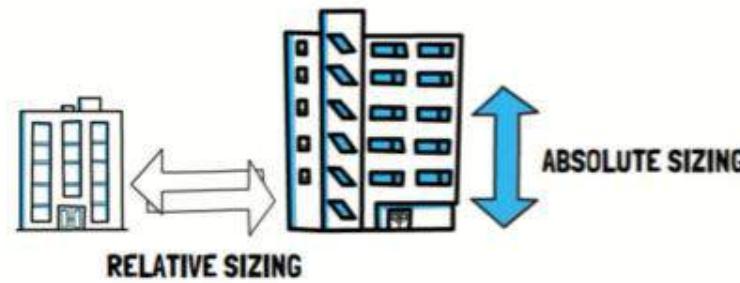
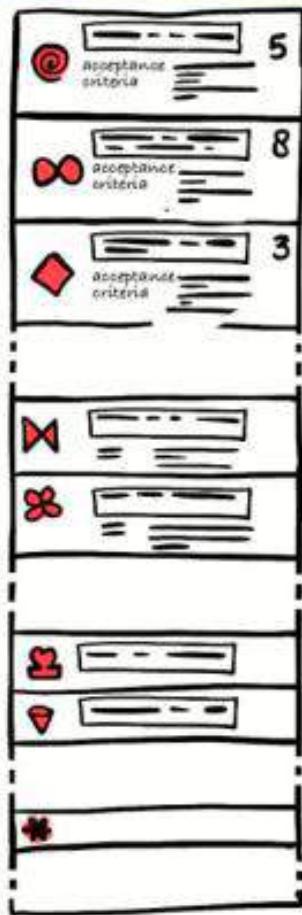
von Product Backlog
Items (PBI's)



SCRUM ARTEFAKTE

PRODUCT BACKLOG / SIZING (AUFWAND)

Product Backlog



PLANNING POKER

0, 1, 2, 3, 5, 8, 13, 20, 40 AND 100.



GRUNDLAGEN FÜR DIE SCHÄTZUNG



Nur das Entwicklungsteam schätzt!

Schätzungen müssen von jenen Personen vorgenommen werden, die auch tatsächlich für die Umsetzung verantwortlich sind und hierfür auch entsprechende Zusagen treffen. Alle anderen Schätzungen sind nicht zulässig!



Alle Schätzungen erfolgen mit relativen Werten

Relative Schätzungen vereinfachen und beschleunigen die Schätzverfahren, bei weiterhin belastbaren Ergebnissen mit geringer Fehlertoleranz.



User Stories werden in "Story Points" geschätzt

Story Points sind ein Instrument zur relativen Schätzung von Größe und Komplexität einer User Story, die über eine reine Betrachtung der Umsetzungszeit geht. In die Bewertung nach Story Points können u.a. die Komplexität, die Menge an Aufgaben, mögliche Risiken und Unsicherheiten mit einfließen.

VORTEIL RELATIVER SCHÄTZUNGEN

Relative Schätzungen

Anwendungsbeispiele zur Veranschaulichung relativer Schätzungen

Münzen



Es ist leichter die relative Anzahl von Münzen in den einzelnen Haufen miteinander zu vergleichen, als die exakte Anzahl zu bestimmen. Innerhalb kürzester Zeit lässt sich abschätzen, dass im Haufen „4“ ungefähr halb so viele Münzen liegen, wie im Haufen mit der Ordnungsnummer „5“.

Länder



Es ist leichter die relative Anzahl von Einwohnern in den einzelnen Ländern zu bestimmen, als deren konkrete Anzahl zu benennen. Wichtig ist hierbei ein klares Verständnis darüber, was tatsächlich der Schätzwert repräsentieren soll (bspw. Schätzung der Einwohnerzahl vs. Schätzung der Fläche)

T-Shirt Sizing



Als Alternative zur Fibonacci Folge wird auch gerne auf das „T-Shirt Sizing“ zurückgegriffen. Das Prinzip bleibt auch hier gleich und soll die Schätzung vereinfachen. Ein T-Shirt der Größe „M“ entspricht hierbei ca. 2x der Größe „S“, welches wiederum ca. 2x der Größe „XS“ entspricht

RELATIVE SCHÄTZUNGEN

Relative Schätzungen erleichtern und beschleunigen den Prozess bei weiterhin belastbaren Ergebnissen

- Relative Schätzwerte bleiben für das Projekt nutzbar, wenn das Projektteam im Projektverlauf „lernt“ und „schneller“ in der Umsetzung einzelner Anforderungen wird
- Benötigen zu Beginn einen Referenzwert. Dieser sollte von einer allgemein verständlichen und gut einschätzbaren User Story ausgehen und muss initial bestimmt werden
- Erlauben eine direkte Vergleichbarkeit zwischen User Stories (eine User Story mit 5 Story Points ist ungefähr doppelt so aufwändig, wie eine User Story mit 3 Story Points)
- Mit Hilfe einer modifizierten Fibonacci Folge, lassen sich Schätzungen schnell durchführen, ohne sich im Detail zu verlieren. Ungenauigkeiten werden bewusst in Kauf genommen, Erfahrungen zeigen, dass hierbei die Genauigkeit insgesamt aber kaum leidet

RELATIVE SCHÄTZUNGEN

Mit Hilfe einer modifizierten Fibonacci Folge, lassen sich Schätzungen schnell durchführen, ohne sich im Detail zu verlieren. Ungenauigkeiten werden bewusst in Kauf genommen, Erfahrungen zeigen, dass hierbei die Genauigkeit insgesamt aber kaum leidet

Fibonacci Folge (modifiziert)

1 2 3 5 8 13 20 40 100

Die Idee hinter der modifizierten Fibonacci Folge als Grundlage für Schätzwerte ist, dass mit zunehmender Größe und Komplexität die Genauigkeit der Schätzung abnimmt und die Unsicherheit steigt. In der Praxis zeigt sich, dass die Werte ab und einschl. 13 –bzw. 20 eher für Epics verwendet werden

STORY POINTS VS. STUNDEN

Frage

„Die **Fibonacci** Folgen ist mir ein Begriff und verstehe auch den Sinn dahinter. Was ich aber nicht verstehe ist die Einheitenbeschreibung von **Story-Points/User Points/T-Shirt Größen?**! Ist es am Ende nicht vollkommen egal ob ich sage ich habe x-Stunden oder x-Story Points im Sprint als Ressourcen zur Verfügung?“

Story Points beschreiben die Komplexität und Struktur einer Anforderung, nicht unbedingt die erforderliche Zeit zur Umsetzung
(relativer Wert!)

Story Points dürfen nicht mit der Teamkapazität verwechselt werden. Diese wird in Stunden berechnet. Beide Einheiten werden später jedoch für Prognosen kommender Sprints benötigt
(exakter Wert in Std.)

VORTEIL VON STORY POINTS AN EINEM BEISPIEL

Sprint 1	
Team Kapazität	10 Stunden
User Story A	5 Story Points
User Story B	3 Story Points
User Story C	5 Story Points
Summe Story Points	13 Story Points
Story Points pro Arbeitsstunde	1,3 Story Points (13/10)

Retrospektive und Einleitung von Prozessverbesserungen

Sprint 2	
Team Kapazität	10 Stunden
User Story D	8 Story Points
User Story E	5 Story Points
User Story F	3 Story Points
User Story G	5 Story Points
Summe Story Points	21 Story Points
Story Points pro Arbeitsstunde	2,1 (21/10)

Team konnte in Sprint 2 deutlich mehr Anforderungen umsetzen
Die wachsende Geschwindigkeit eines Teams verändert nicht die Größe der Story!

Schätzverfahren Planning Poker

Planning Poker eignet sich gut zur Vorbereitung von Sprint - Plannings

Was ist es?

Planning Poker ist eine der gängigsten und besten Methoden zur konkreten Schätzung einzelner User-Stories Anhand von „Poker-Karten“, welche die Fibonacci Folge abbilden können Schätzungen vorgenommen werden

Wann einsetzbar?

Planning Poker bietet sich ebenfalls für Backlog Refinements, oder zur Vorbereitung von Sprint-Plannings ein um eine belastbare Sprint Planung aufzustellen. Die Schätzungen sollten jedoch nicht im Sprint-Planning selbst erfolgen.

SCHÄTZVERFAHREN PLANNING POKER



PLANNING POKER

BEDEUTUNG DER KARTEN

- 0 bedeutet, dass der Schätzende die Aufgabe für zu klein hält. Die Aufgabe sollte also mit anderen zusammengelegt werden.
- Die Zahlen (außer der 0) stehen für einen konkret geschätzten Aufwand (zur Einheit siehe unten).
- ∞ als Zeichen für „unendlich“ bedeutet, dass der Schätzende die Aufgabe für zu groß hält. Sie sollte also in mehrere Aufgaben aufgeteilt werden.
- ? bedeutet, dass man die Aufgabe nicht ausreichend verstanden hat. Sie muss also noch einmal genauer erklärt werden.
- Hält jemand die Karte mit der Kaffeetasse hoch, schlägt er eine Pause vor.

PLANNING POKER

ABLAUF EINER SITZUNG

Ablauf einer Sitzung

Eine Planning-Poker-Sitzung läuft so ab:

Jedes Mitglied des Schätzteams erhält einen kompletten Kartensatz. Der Sitzungsleiter (häufig der Projektleiter) stellt eine Aufgabe vor und bittet dann um die Schätzungen dazu. Alle Teilnehmer bekommen kurz Zeit, um sich für eine Karte zu entscheiden. Dann halten alle gleichzeitig die von ihnen gewählte Karte hoch.

Sehr wichtig ist, dass alle gleichzeitig die von ihnen gewählten Karten hochhalten. Damit wird verhindert, dass die Teilnehmer sich bewusst oder unbewusst einer zuerst genannten Schätzung anpassen. Dieser Anpassungseffekt ist in der Psychologie unter den Stichworten „Priming“ oder „Ankern“ wohlbekannt und führt beim Planning Poker zu weniger realistischen Schätzungen.

PLANNING POKER

ABLAUF EINER SITZUNG

Vorbereitung

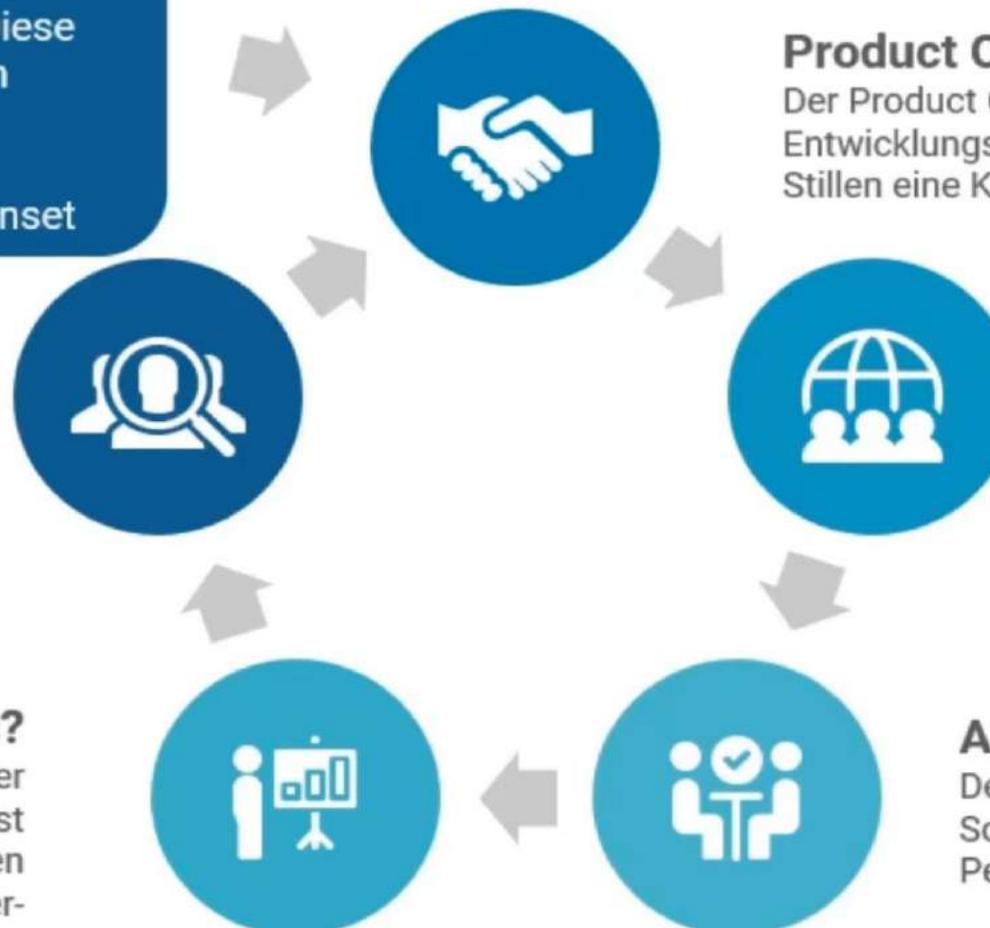
- Vor der ersten Schätzung sollte eine Referenzstory festgelegt werden. Diese wurden im besten Falle bereits vom Team abgeschlossen
- Jeder Entwickler benötigt ein vollständiges Planning Poker Kartenset

Unstimmigkeit

Ergibt sich kein Konsens, so wird eine weitere Schätzrunde durchgeführt. In der Regel findet sich nach max. 3 Runden ein Konsens, ansonsten muss die User-Story ggfs. zurückgestellt werden

Konsens?

Ergibt sich nach der Erläuterung ein Konsens, so ist das Ergebnis zu dokumentieren und mit der nächsten User-Story fortzufahren



Product Owner stellt User Story vor

Der Product Owner erklärt die User-Story, das Entwicklungsteam stellt Rückfragen und wählt im Stillen eine Karte mit entsprechendem Schätzwert aus

Gleichzeitiges Aufdecken

Das Entwicklungsteam deckt ohne vorherige Abstimmung gleichzeitig die Karten auf

Analyse der Schätzung

Der jeweils höchste und niedrigste Schätzwert werden von den jeweiligen Personen kurz erläutert

PLANNING POKER

BEISPIEL

Beispiel:



Ein Entwicklungsteam schätzt mittels Planning Poker den Aufwand für die Realisierung eines neuen Ablaufs in einem Webshop. In der ersten Schätzrunde gibt es einen Ausreißer nach unten. Auf Nachfrage begründet der entsprechende Entwickler seine Schätzung damit, dass ein ähnlicher Prozess bereits für einen anderen Kunden realisiert wurde. Nach kurzer Diskussion gibt es eine zweite Schätzrunde. Dabei liegen die Schätzungen näher zusammen und sind im Mittel geringer als in der vorherigen Runde.