

SCRUM

Foundation



Herzlich willkommen zur Scrum Foundation Schulung!

Organisatorisches

- Vorstellungsrunde
- Agenda
- Einordnung der Schulung

Wer ist das mITSM?

- Gegründet: 2003
- Training und Personenzertifizierungen
 - IT Service Management
 - Information Security Management
 - Projektmanagement
 - Datenschutz
 - Mimikresonanz
 - ITSec, Blockchain, Requirements Engineering, ...
- Zusätzlich bieten wir
 - Beratung
 - Coaching
 - Auditierung



TRAINING ORGANIZATION
ACCREDITED BY PeopleCert



Kooperationspartner von



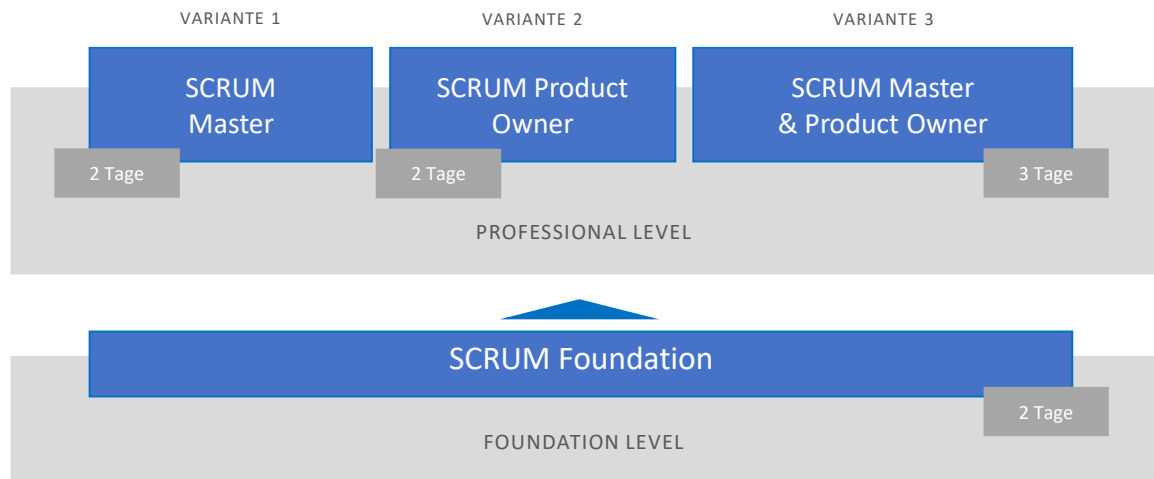
Vorstellungsrunde

- Name
- Firma und Verantwortung
- Erfahrung mit SCRUM
- Erfahrungen mit Projektmanagement / weiteren agilen Methoden
- Erwartung an die Schulung
- Eventuelle konkrete Projekte, bei denen SCRUM eingesetzt werden soll

Einordnung der Schulung / Kursnutzen

- Grundlegendes Verständnis von SCRUM
- Anwendungsmöglichkeiten in der Praxis
- Verstehen der Konzepte und Prinzipien hinter SCRUM
- Kenntnisse über die wichtigsten Bestandteile von SCRUM
- Vorbereitung auf die Foundation Prüfung

SCRUM Schulungs- und Zertifizierungsschema



Prüfung „ITEMO SCRUM Foundation“

- Sprache: Deutsch / Englisch
- Dauer: 45 Minuten
- Closed Book, d.h. keine Hilfsmittel zulässig
- 30 Multiple Choice-Fragen
- **Eine** von **vier** Antwortmöglichkeiten ist korrekt
- Zum Bestehen der Prüfung werden 20 der 30 Punkte benötigt

☒	☐	☐	☐
☐	☒	☐	☐

Agenda

- Motivation & Hintergrund von SCRUM und agilen Methoden
- SCRUM im Überblick: Rollen, Rituale (Ereignisse), Artefakte
- SCRUM Rollen
- Requirements Engineering in SCRUM
- Sprints
- Fortschrittsmessung (Metriken)
- Release Management in SCRUM
- Große und verteilte Projekte
- Weitere Themen
- Prüfungsvorbereitung & Prüfung



Grundlagen & Motivation

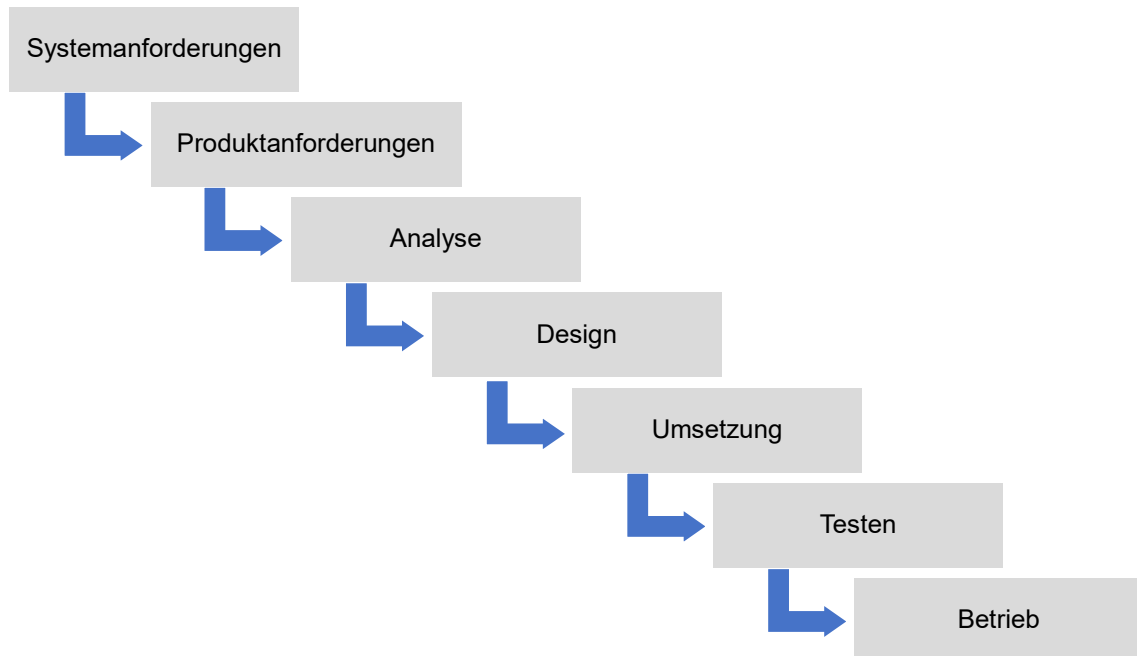
- Ausgangssituation
- Zusammenhang mit anderen Normen / Frameworks, etc.
- Wichtige Grundbegriffe

CHAOS Report „agile versus waterfall“

Größe	Methode	erfolgreich	überzogen	fehlgeschlagen
Alle	Agil	39%	52%	9%
	Wasserfall	11%	60%	29%
Groß	Agil	18%	59%	23%
	Wasserfall	3%	55%	42%
Mittel	Agil	27%	62%	11%
	Wasserfall	7%	68%	25%
Klein	Agil	58%	38%	4%
	Wasserfall	44%	45%	11%

Vgl. CHAOS Report, Standish Group; Zusammengefasst alle Softwareprojekte von 2011 – 2015 in der CHAOS Datenbank (über 10.000), aufgeteilt in agile Prozesse und Wasserfallmethoden.

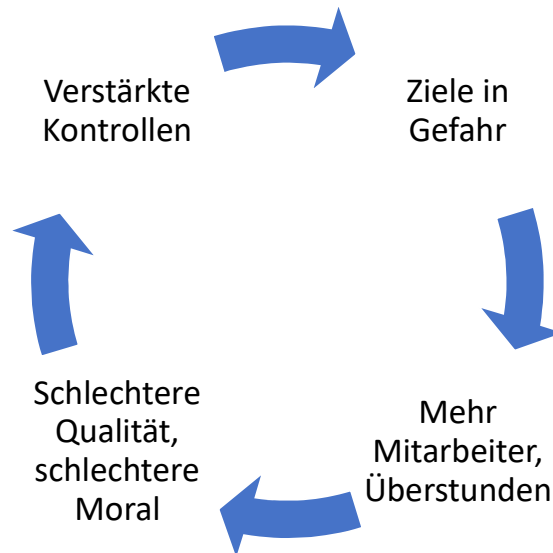
Klassisches Projektmanagement – Das Wasserfall-Modell



Nachteile:

- Dauer der einzelnen Phasen unklar
- Eine Phase darf erst begonnen werden, wenn die vorherige abgeschlossen ist
- Bei Änderungen muss in die vorherige Phase zurück gesprungen werden
- Spätes Feedback erst am Schluss
- Overhead durch Planung und Reporting

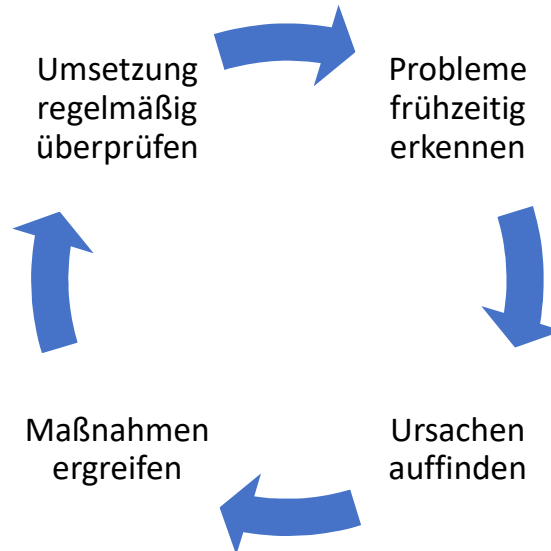
Der Teufelskreislauf klassischer Methoden



Unterbrechen des Teufelskreislafs

- Traditionelle Vorgehensweisen ...
 - ... versuchen, alle möglichen Eventualitäten zu antizipieren
 - ... versuchen, zu Beginn bereits alle Arbeitsdetails einzuplanen
 - ... erhalten erst sehr spät Rückmeldung über den Fortschritt
- Ziel agiler Methoden ist es ...
 - ... möglichst früh Rückmeldung über den Projektfortschritt zu erhalten
 - ... Probleme und Hindernisse möglichst früh zu erkennen
 - ... möglichst früh entsprechende Gegenmaßnahmen zu ergreifen

Der empirische Kreislauf agiler Methoden



Empirische Prozesssteuerung basiert auf den drei Säulen Transparenz, Überprüfung und Anpassung

Das agile Manifest

- **Individuen und Interaktionen** sind wichtiger als Prozesse und Tools
- **Funktionierende Software** ist wichtiger als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** ist wichtiger als Vertragsverhandlungen
- **Reagieren auf Veränderung** ist wichtiger als das Befolgen eines Plans

*„We are uncovering better ways of developing software by
doing it and helping others do it.“*

agilemanifesto.org

Der Fokus sollte auf den Werten auf der linken Seite liegen. Dies bedeutet nicht, dass die rechten Werte nun unwichtig sind.

<http://agilemanifesto.org/iso/de/manifesto.html>

Nutzen und Eigenschaften agiler Methoden

- Menschen und Kommunikation stehen im Mittelpunkt
- Transparenz
- Flexibilität
- Empirisches Arbeiten
- Kundenorientierung
- Iterativ
- Vertrauen
- Timeboxing
- Fokus auf funktionsfähigem Produkt (Software)
- Reduzieren des Overheads klassischer Projektmanagement-Methoden

Transparenz

- Der aktuelle Projektstand ist immer und für jeden sichtbar
- Probleme und Hindernisse werden offen und ehrlich angesprochen

Flexibilität

- Auf Veränderungen kann schnell reagiert werden

Empirisches Arbeiten

- Die Arbeitsweise wird regelmäßig überprüft und angepasst

Kundenorientierung

- Der Kunde wird von Beginn an in den Entwicklungsprozess mit eingebunden

Iterativ

- Kontinuierliche Verbesserung der eingespielten Verfahren
- Regelmäßige Begutachtung aller Ergebnisse und des Produktes

Vertrauen

- Das Team ist bevollmächtigt und trifft selbständig Entscheidungen
- Stakeholder müssen Vertrauen in die Fähigkeit des Teams haben

Timeboxing

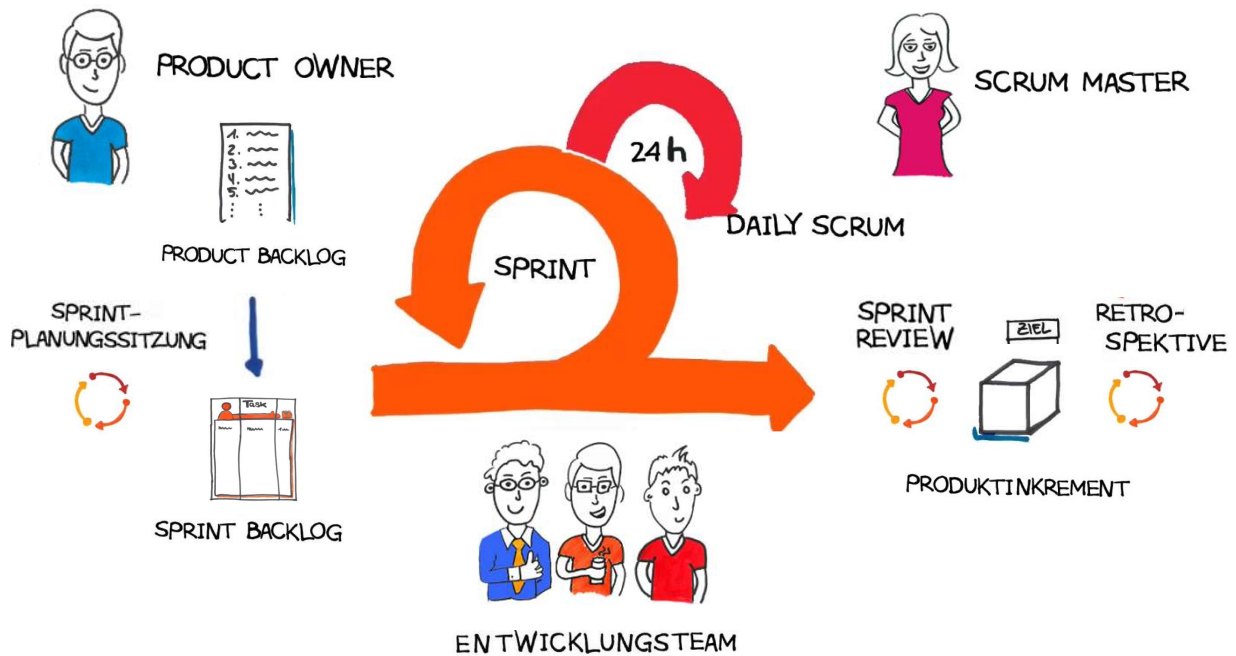
- Absolute Termintreue für feste Ereignisse (maximale Dauer)
- Festgelegte Rituale dürfen eine definierte Dauer nicht überschreiten

Fokus auf funktionsfähigem Produkt (Software)

- Zu möglichst jedem Release-Termin existiert eine funktionsfähige Produktversion

SCRUM im Überblick: Ein agiles Management-Framework



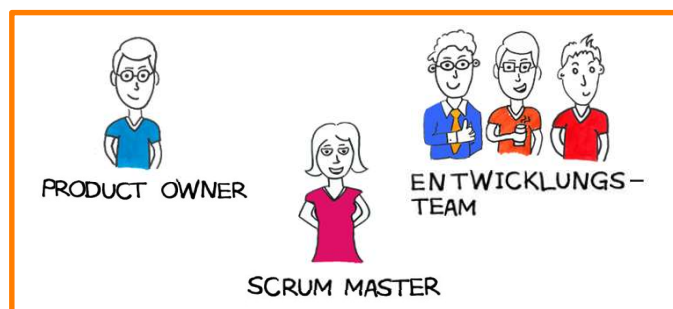


Zentrale Begriffe und Eigenschaften:

- **Product Backlog:** Das Product Backlog enthält als einzige zentrale Quelle alle Anforderungen, die zur Erreichung des Entwicklungsziels umgesetzt oder erbracht werden müssen.
Die Einträge (Product Backlog Items, PBIs) sind priorisiert und ihre Aufwände sind abgeschätzt.
- **Sprint:** Ein Sprint ist ein Arbeitszyklus (eine Iteration), in dem eine Teilmenge der Anforderungen aus dem Product Backlog umgesetzt wird. Er dauert maximal 30 Tage.
- **Produktinkrement:** Das Produktinkrement ist der aktuelle Entwicklungsstand. Es ist immer lauffähig, getestet und dokumentiert und damit theoretisch auslieferbar. Die Anforderungen aus dem Product Backlog werden im Laufe der Entwicklung in das Produktinkrement umgesetzt, d.h. in jedem Sprint wird das Produktinkrement erweitert.

Drei Rollen

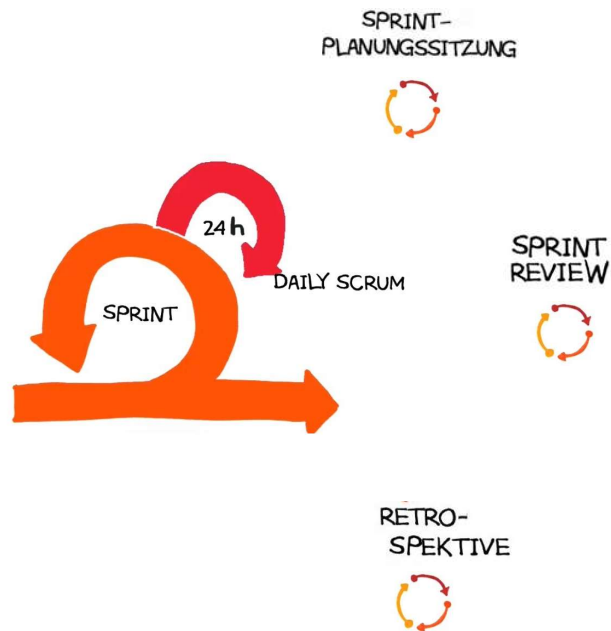
- Product Owner
 - Requirements Engineering, Projekt- / Produktverantwortung
- SCRUM Master
 - Überwachung der Einhaltung von SCRUM
 - Verantwortung für das Beseitigen von Problemen und Hindernissen
- Entwicklungsteam
 - Ausführung der Arbeiten
 - Planung der Sprints



SCRUM Team

Fünf Rituale (Ereignisse)

- Sprint
- Sprint Planning
- Daily SCRUM
- Sprint Review
- Sprint Retrospektive



Sprint

- Gleich lang (2 bis 4 Wochen), maximal 30 Tage

Sprint Planning (Sprint Planungssitzung)

- Zu Beginn des Sprints
- Festlegen der Sprint-Ziele und -Inhalte

Daily SCRUM

- Tägliche kurze Besprechung (max. 15 Min) des Entwicklungsteams
- Was war? Was ist? Was behindert uns?

Sprint Review

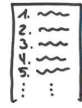
- Vorstellung des Produktinkrements am Ende des Sprints
- Einholen von Feedback der Stakeholder

Sprint Retrospektive

- Kontinuierliche Verbesserung
- Zukünftige Maßnahmen beschließen

Drei Artefakte

- Product Backlog
- Sprint Backlog
- Produktinkrement



PRODUCT BACKLOG



PRODUKTINKREMENT



SPRINT BACKLOG

- Weitere Begriffe
 - Burndown Charts – Fortschrittsmessung, Zielerreichung
 - Velocity – Entwicklungsgeschwindigkeit
 - Definition of Done – gemeinsame Akzeptanzkriterien
 - Impediments Backlog – Backlog für Hindernisse und Störfaktoren

Product Backlog

- Sammlung aller Anforderungen an das Produkt und das Projekt

Sprint Backlog

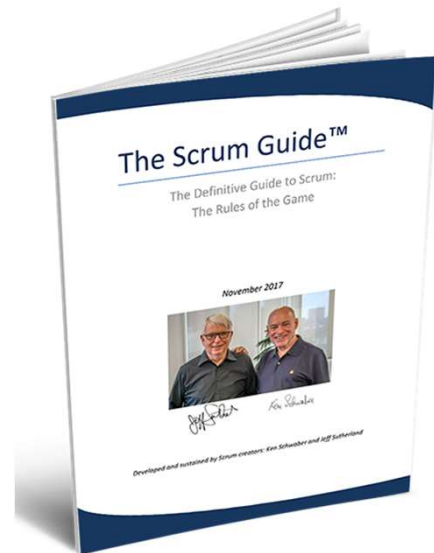
- Auszug des Product Backlogs mit den Anforderungen für einen Sprint

Produktinkrement

- Im Optimalfall: lauffähiger (releasebarer) Entwicklungsstand

SCRUM Werte (Values)

- Mut (Courage)
- Fokus (Focus)
- Selbstverpflichtung (Commitment)
- Respekt (Respect)
- Offenheit (Openness)



Die SCRUM Werte (values) wurden im Juli 2016 zum SCRUM Guide hinzugefügt, um sie noch expliziter zu unterstreichen.

Vgl. <https://blog.scrum.org/updates-scrum-guide-5-scrum-values-take-center-stage/>

Courage

- Transparenz fördern, auch wenn das bedeutet, dass man falsch liegt oder die eigene Meinung nicht der Teammeinung entspricht

Focus

- Fokussiert auf den aktuellen Sprint auf dessen Ziel hinarbeiten

Commitment

- Sich gemeinsam mit dem Team zum Sprint-Ziel verpflichten

Respect

- Anderen zu helfen, die Dinge, in denen man selbst gut ist, zu lernen statt sie dafür zu verurteilen dass sie nicht gut in diesen Dingen sind

Openness

- Hindernisse und Probleme offen ansprechen

Zentrale Eigenschaften und Aspekte

- **Frühzeitiges Erkennen von Problemen**
 - Rückmeldung innerhalb weniger Wochen (2-3 Sprints) über Fortschritt, Probleme und Hindernisse
 - Dadurch: frühzeitiges Ergreifen von Gegenmaßnahmen
- **Mitarbeitermotivation und Mitarbeiterzufriedenheit** durch
 - Bevollmächtigung der Mitarbeiter
 - Selbstorganisation
- **Kundenzufriedenheit**
 - Enge Zusammenarbeit mit dem Kunden
 - Einbeziehung in den Entwicklungsprozess
 - Wirtschaftlicher Nutzen
 - Time-to-Market: Priorisierung und geringerer Overhead
 - Fokus auf Produktqualität

Wann ist SCRUM nicht die geeignete Lösung?

- Wenn man SCRUM anwendet, nur weil es jeder macht.
- Wenn man die Kontrolle nicht an das Team übertragen kann oder will.
- Wenn man keinen Wandel im Unternehmen durchsetzen kann oder will.
- Wenn das Business oder das Management nicht voll dahinter steht.
- Wenn man SCRUM ohne Experten einführt.

SCRUM bedeutet einen kulturellen Wandel und erfordert eine Änderung in der Unternehmensstruktur.

SCRUM funktioniert nur, wenn dieser Wandel auch gewollt und durch das Management vertreten wird.

SCRUM Rollen

- Product Owner
- Entwicklungsteam
- SCRUM Master

Aufgaben des Product Owners



- Repräsentation der Stakeholder und derer Bedürfnisse
 - Verstehen des Mehrwerts aus Kundensicht
 - Enge und regelmäßige Abstimmung mit dem Kunden (Auftraggeber)
 - Kommunikation der Produktvision
 - Brücke zwischen Kunde (Auftraggeber) und SCRUM Team

- Enge Zusammenarbeit mit dem Entwicklungsteam
 - Hilfe beim Verständnis von Anforderungen
 - Überprüfung und Abnahme der Arbeitsergebnisse

- Verantwortung für das Entwicklungsziel und den Projekterfolg

Aufgaben des Product Owners



- Erfassung und Beschreibung der Anforderungen
 - Erstellung von Produktkonzept und Product Backlog
 - Kontinuierliche Bearbeitung (Verfeinerung und Priorisierung des Product Backlogs)
 - Setzen der Status „ready“ (umsetzungsbereit) und „done“ (abgenommen)
- Produkt- und Projektmanagementaufgaben
- Entscheidung über
 - Auslieferungszeitpunkte, Funktionalitäten, Kosten
 - Sprint-Abbruch bei geänderten Rahmenbedingungen

Empfehlungen und bewährte Praktiken



- Arbeit des Product Owners ist in der Regel eine Vollzeitaufgabe (wird oft unterschätzt)
- Projekterfolg leidet, wenn die Rolle des Product Owners nicht qualifiziert besetzt wird:
 - Wichtigste Eigenschaft: Verstehen der Kundenbedürfnisse und Vermittlung an das Team
 - Produktmanager oder Marketingmitarbeiter häufig in dieser Rolle
- Häufigste Fehler: Product Owner ist
 - überlastet
 - nicht bevollmächtigt

Empfehlungen und bewährte Praktiken



- Ausreichend Zeit mit dem Team verbringen:
 - Faustformel: Mindestens eine Stunde täglich
 - Teilnahme an SCRUM-Besprechungen

- Regelmäßig Abstimmung mit z.B. Marketing, Vertrieb, Service oder IT (Anforderungswshops)

- Die Abnahme von Anforderungen (Product Backlog Items) kann jederzeit während des Sprints erfolgen. Die Abnahme sollte in jedem Fall aber VOR dem Review erfolgen

Die Abnahme von Anforderungen bereits während des Sprints hat mehrere Vorteile:

- Der PO weiß durch die regelmäßige Abnahme wo das Projekt steht und welche Anforderungen noch nicht fertiggestellt sind.
- Die Timebox des Sprint Reviews kann so eingehalten werden, wenn sich der PO im Vorfeld bereits einen Überblick verschafft hat.
- Das Sprint Review kann vorrangig genutzt werden, um den aktuellen Entwicklungsstand zu präsentieren und Feedback der Teilnehmer einzuholen

Aufgaben des Entwicklungsteams

- Umsetzung der Anforderungen
- Entscheidung über die Anforderungen im nächsten Sprint (Sprint Backlog)
- Selbständige Team-Organisation
 - Festlegung der einzelnen Arbeitsschritte (Tasks) zur Umsetzung der Anforderungen und Aufteilung an die Team-Mitglieder
 - Festlegung von Arbeitsweisen, Richtlinien, Werkzeugen etc.
 - Disziplinierung durch Sprint Backlog und Daily SCRUM
- Unterstützung des Product Owners durch Bewertung und Abschätzung der Anforderungen im Product Backlog



Eigenschaften des Entwicklungsteams

- Interdisziplinäre Besetzung (Cross-Skilling)
- Kleine Größe (3-9 Mitglieder):
 - mit Vollzeitmitgliedern besetzt
 - auch Teilzeitmitglieder möglich
- Hohes Maß an Kooperation und gegenseitiger Zusammenarbeit erforderlich
- Teambuilding (nach Tuckman):



Nach dem Tuckman Phasenmodell entsteht ein Team nicht über Nacht, sondern wächst langsam zusammen (mindestens 2-3 Sprints):

- Es muss zunächst gebildet werden (forming)
- Danach muss es eine Auseinandersetzungsphase durchlaufen (storming)
- Anschließend muss es sich auf gemeinsame Arbeitsweisen und Normen einigen (norming)
- Erst dann kann es produktiv arbeiten (performing)

Aufgaben des SCRUM Masters



- Etablierung des SCRUM-Prozesses und Achtung auf Einhaltung der SCRUM-Regeln
 - Förderung neuer Denk- und Verhaltensweisen
 - Aufdecken alter, kontraproduktiver Gewohnheiten
 - Unterstützung der Organisation bei der Beseitigung von Hindernissen
 - Schutz und Unterstützung des Entwicklungsteams

- Sicherstellung der Zusammenarbeit zwischen PO und EWT

- Hilfe und Unterstützung bei der Einführung von SCRUM-Praktiken

Der SCRUM Master als Coach und Change Agent



- Der ideale SCRUM Master ...
 - ... steht auch in schwierigen Situationen dem Entwicklungsteam bei und achtet auf die Einhaltung von SCRUM
 - ... kann gut moderieren
 - ... hat fachliche Erfahrung im Projektbereich

- Im Idealfall hat der SCRUM Master irgendwann (fast) nichts mehr zu tun
 - Alle kennen die Regeln und achten selbst auf deren Einhaltung
 - Alle erscheinen pünktlich zu den Meetings
 - Die Rolle des SCRUM Masters wird dann auch oft von einem Mitglied des Entwicklungsteams mit übernommen

Empfehlungen und Good Practices



- Kein Teamleiter: keine Autorität bezüglich der Arbeitsorganisation im SCRUM Team
- Akzeptanz durch das restliche SCRUM Team:
→ Einbeziehung in den Auswahlprozess (besonders bei externem SCRUM Master)
- Keine Personalverantwortung für die Teammitglieder ausüben:
→ Beeinträchtigung von Offenheit und Kommunikation im Scrum Team

Die Rolle des SCRUM Masters ist nicht mit der Rolle eines traditionellen Projektleiters gleichzusetzen



Requirements Engineering in SCRUM

- Das Product Backlog
- Anforderungsbeschreibung
- User Stories
- Features
- Priorisierung
- Aufwandsschätzungen

Das Product Backlog

- Zentrales Instrument zum Erfassen und Managen von Anforderungen bzw. Product Backlog Items (PBIs), z.B.:
 - Funktionale und nicht-funktionale Anforderungen
 - Anforderungen an externe Schnittstellen
 - Andere Aufgaben im Projekt (z. B. Beseitigen von Defekten / Bugs)
 - Änderungsanträge (Change Requests)
 - Umsetzung von Verbesserungsmaßnahmen
 - Berücksichtigung von Risiken und Abhängigkeiten

- Lebendes Dokument
 - Kontinuierliche Veränderung und Aktualisierung
 - ist (vor allem zu Beginn eines Projekts) nie vollständig

Das Product Backlog

Beispiel für ein einfaches Product Backlog:

ID	Anforderung	Akzeptanzkriterien	Priorität	Aufwands- abschätzung	Status

- Sortierung von Einträgen nach Priorität (absteigend)
- Verwendung unterschiedlicher Detaillierungsstufen
 - Detaillierte Spezifikationen für hoch priorisierte oder riskante Anforderungen

Empfehlungen für Status:

- Ready: Eine Anforderung ist umsetzungsbereit
- Done: Eine Anforderung ist umgesetzt, genügt somit den Akzeptanzkriterien und der Definition of Done
- Rejected: Eine Anforderung wird zurückgestellt und ist im Moment nicht relevant

Das Product Backlog

- Alle Einträge sind priorisiert:
 - abhängig vom Kundennutzen (Product Owner)
 - der Bewertung technischer Risiken/Machbarkeit (Entwicklungsteam)
- Einträge sind abgeschätzt (Aufwand)
 - Zukünftige Anforderungen immer abgeschätzt
 - Anmerkung: Bei unklaren Anforderungen ist die Schätzung entsprechend ungenau

Ohne vollständig abgeschätztes Product Backlog ist keine Releaseplanung und keine Fortschrittsmessung möglich



Das Product Backlog

- Kategorisierung von Anforderungen in verschiedene Phasen der Spezifikation, z.B.:
 - Idee
 - Anforderung
 - Spezifikation 

- Gruppierung zu Themenblöcken, z. B.
 - Features
 - Epics 

Der Product Owner trägt die Verantwortung für das Produkt Backlog Management



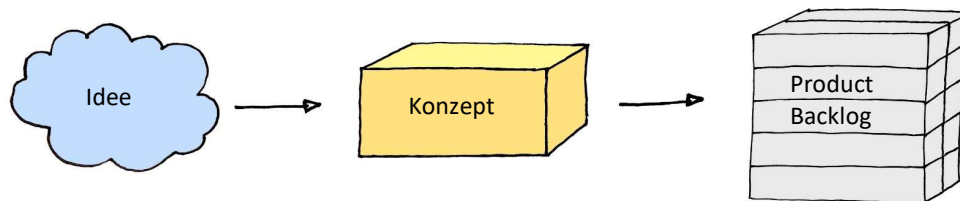
Der Product Owner kann das Recht erteilen, sodass auch andere Personen Einträge im Product Backlog erstellen können. Er ist allerdings immer final verantwortlich.

Es ist gute Praxis, dass der Product Owner dieses Recht delegiert, nicht aber ein Automatismus. Das Product Backlog ist sein Artefakt, so wie das Sprint Backlog das Artefakt des Teams ist.

Produktkonzept

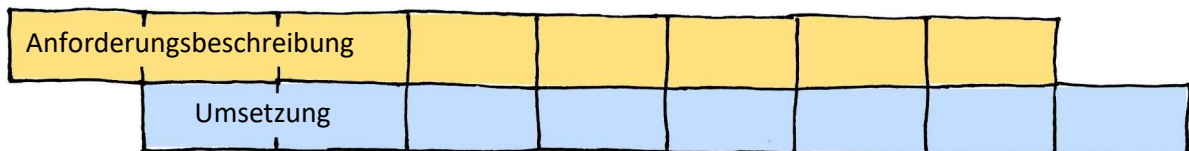
Oft ist es sinnvoll, vor dem Product Backlog ein Produktkonzept zu erstellen:

- Inhalt: wesentliche Leistungsmerkmale und Zielgrößen der Produktidee
 - Kurz und knapp, nur das Wesentliche
 - Fokus auf den Mehrwert (Business Value aus Sicht des Kunden)
- Vorteile
 - Eruierung des Marktpotentials
 - Grundlage für eine effiziente Erstellung des Product Backlogs



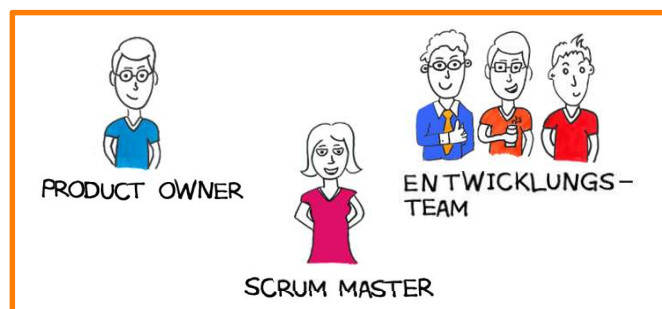
Anforderungsbeschreibung in SCRUM

- Erfassung der Anforderungen iterativ und inkrementell im Laufe des Projekts parallel zur Entwicklung und Umsetzung:
 - Anforderungsbeschreibung und Umsetzung zeitnah
 - Vermeidung unnötiger Überproduktion
 - Besseres Verständnis für Anforderungen aufgrund des aktuellen Entwicklungsstands auf allen Seiten
 - Vermeidung von unausgeglichem Arbeitsanfall in der Entwicklung, da nicht mehr alle Anforderungen auf einmal übergeben werden



Product Backlog Refinement

- Ziel: gemeinsames Verständnis über (auch neue) PBIs , die in den folgenden Sprints anstehen
- Aufgaben:
 - Detaillieren
 - Schätzen
 - Aufteilen und Ordnen von PBIs
- Zeit: keine feste Vorgabe, max. 10% der Kapazität des Entwicklungsteams



Das Product Backlog Refinement findet als kontinuierlicher Prozess statt. Hierbei besprechen und diskutieren Product Owner und Entwicklungsteam neu hinzugekommene Anforderungen. Es erfolgt bereits eine erste Abschätzung durch das Entwicklungsteam. Das Refinement umfasst sowohl die Detaillierung von Anforderungen als auch das Aufsplitten von Anforderungen.

Wichtige Tipps für das Refinement:

- Immer vorbereitet in ein Refinement gehen
- Das Team sollte sich für jedes zu besprechende Item eine Timebox setzen
- Das Team sollte im Refinement nicht versuchen, bereits konkrete Lösungen auf Taskebene zu finden
- Es muss ebenfalls im gesamten Scrum-Team ein einheitliches Verständnis zu dem Zustand von „Ready“ für eine User Story geben
- Schätzungen können, je nachdem ob ein Epic, Feature oder eine User Story besprochen wird, mit unterschiedlichen Herangehensweisen erfolgen

Inkrementelle Innovation

- „Step by Step“
- Stetige und schrittweise Verbesserung von bestehenden Produkten
- Anforderungen pro neuer Version möglichst minimal
- Identifikation der kleinsten Menge an sinnvoll releasebaren Merkmalen für jede Version (Verbesserung des Vorgängermodells)
- Schnelle und häufige Auslieferung neuer Funktionalität
- Reduzierung der Time-to-Market

Im Optimalfall liefert jeder Sprint in SCRUM eine potentiell nützliche bzw. nutzbare Version des Produkts

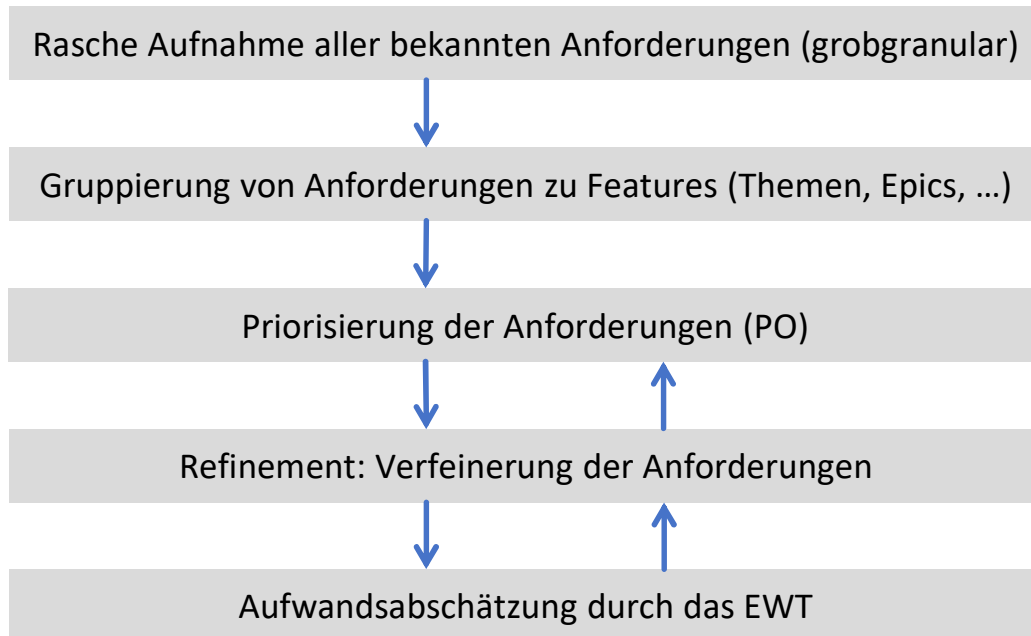


Inkrementelle Innovation ist ein zentrales Konzept von SCRUM und meist das Ergebnis eines kontinuierlichen Verbesserungsprozesses.

Zusätzliche Vorteile der inkrementellen Innovation:

- Schnellerer Break-even
- Schnellere Rückmeldung und bessere Vorhersage der Kundenbedürfnisse
- Kurze Releasezyklen erleichtern die Abschätzung der Zielerreichung

Pflege des Product Backlogs

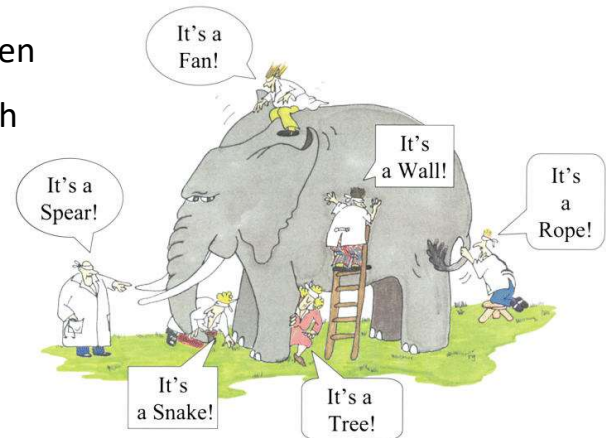


Hinweis: Anforderungen können auch während des Projektverlaufs wieder aus dem Product Backlog entfernt oder zurückgestellt (z.B. Status „rejected“) werden

User Stories

Als **<Rolle>** möchte ich **<Ziel>**, um **<Nutzen>**

- Beschreibung von Anforderungen aus verschiedenen Rollensichten (Stakeholder, Anwender)
- Formulierung in Alltagssprache, ca. 2-3 Sätze
- Jede User Story hat Akzeptanzkriterien
- Nutzen jeder User Story ist ersichtlich
- User Stories sind so formuliert und strukturiert, dass der Blick fürs Gesamtprojekt nicht verloren geht



Vgl. <http://www.agilebuddha.com/agile/user-stories-lack-of-big-picture-leads-to-blind-man-product>

Beispiel: User Story und Akzeptanzkriterien

Titel: Neu-Kunde registrieren

Story ID: 4711

User Story

Als **Neu-Kunde** möchte ich mich
im **Onlineshop registrieren**,
um **Bestellungen aufgeben zu können**.

Datum

09.01.2019

Autor

M. Mustermann

Akzeptanzkriterien

Die Registrierung ist abgeschlossen, wenn der
Kunde eine systemseitige Bestätigungsmail
erhalten hat.

Wichtigkeit

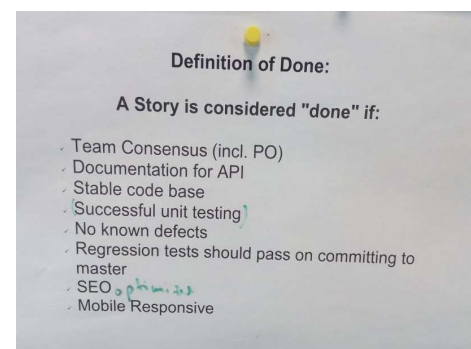
hoch

Schätzung

5

Definition of Done (DoD)

- Kriterien, die (zusätzlich) erfüllt sein müssen, damit PBIs als fertig („Done“) gelten
- Sorgt für ein gemeinsames Verständnis innerhalb des Scrum Teams (Commitment)
- Die DoD kann z.B. in Retrospektiven angepasst werden
- Tipps
 - evtl. „Level of Done“ für verschiedene Fertigstellungsebenen verwenden (z.B. für Story, Sprint, Release)
 - Zusätzliche Beschreibung von Rahmenbedingungen



Bildquelle: <https://www.scrum-academy.de/product-owner/wissen/definition-of-done-simpel-und-doch-komplex/>
 Beispiel von <https://www.101ways.com/definition-of-done-10-point-checklist/>

Definition of DONE! 10 Point Checklist

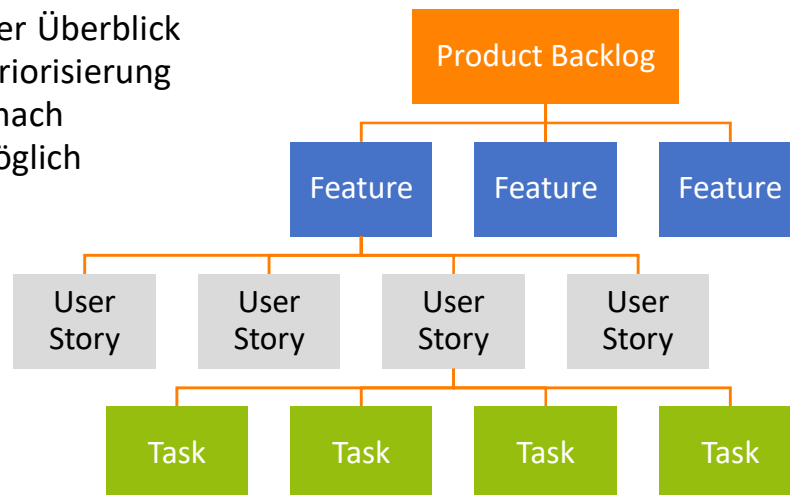
A key principle of agile software development is “done means DONE!”

To be more specific, here’s a 10 point checklist of what constitutes ‘feature complete’...

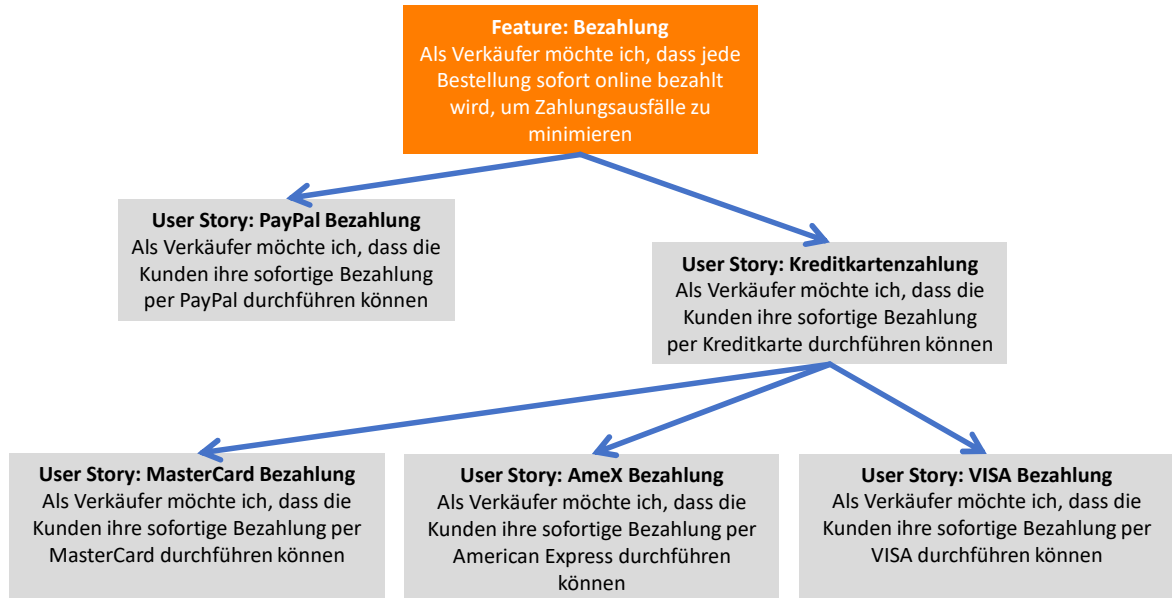
1. Code produced (all "to do" items in code completed)
2. Code commented, checked in and run against current version in source control
3. Peer reviewed (or produced with pair programming) and meeting development standards
4. Builds without errors
5. Unit tests written and passing
6. Deployed to system test environment and passed system tests
7. Passed UAT (User Acceptance Testing) and signed off as meeting requirements
8. Any build/deployment/configuration changes implemented/documented/communicated
9. Relevant documentation/diagrams produced and/or updated
10. Remaining hours for task set to zero and task closed

Features

- Gruppierung von PBIs zu Features
 - Synonyme: Themen, Module, Epics
- Dadurch:
 - verbesserter Überblick
 - leichtere Priorisierung
 - Abnahme nach Feature möglich

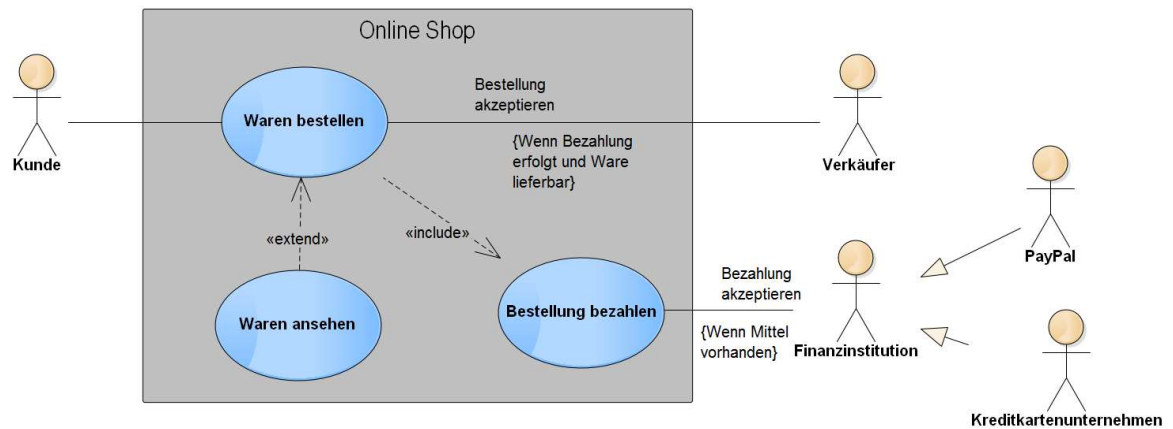


Beispiel: Feature und aufgeteilte User Stories



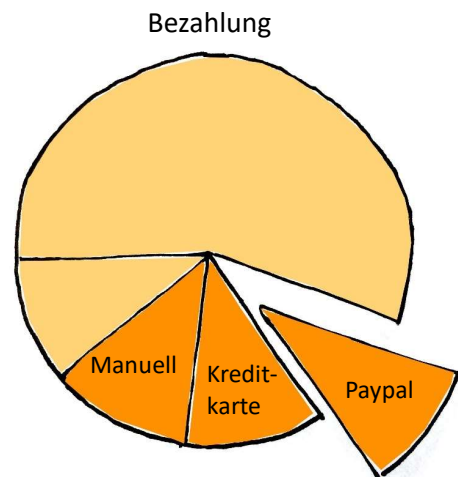
Features & Use Cases

Beispiel für ein UML Use Case Diagramm



Aufteilen von User Stories (Splitting Stories)

- Eine User Story sollte in einem Sprint umgesetzt werden können
- Der Nutzen für das Business muss am Ende erhalten bleiben, wenn man große User Stories in kleinere unterteilt
- Möglichkeiten der Unterteilung
 - Horizontale Unterteilung
(technische Ebenen, Komponenten)
 - Vertikale Unterteilung
(funktionale Ebenen, Features)
- Vertikale Unterteilung in Scrum bevorzugt



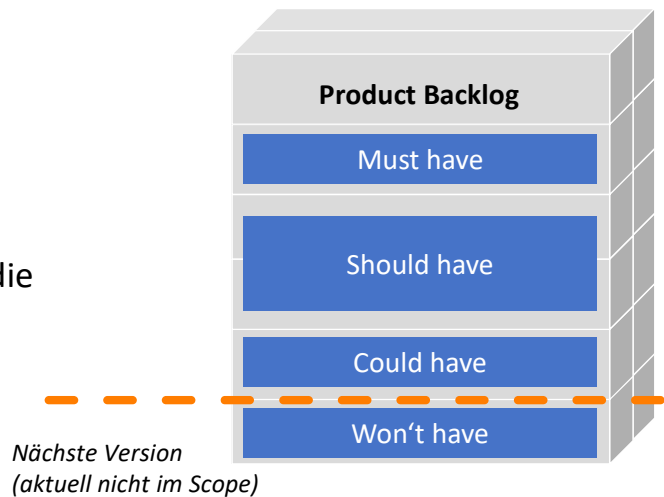
Grundsatz beim Unterteilen guter Anforderungen

Anforderungen sollten die INVEST-Eigenschaften erfüllen:

- **Independent** (Unabhängig): In beliebiger Reihenfolge umsetzbar
- **Negotiable** (Verhandelbar): Verhandlungsspielraum für das Sprint-Planning, um maximale Produktivität zu erzielen
- **Valuable** (Nützlich): Der Nutzen muss klar erkennbar sein
- **Estimatable** (Schätzbar): Anforderungen sind klar formuliert und werden vom Entwicklungsteam verstanden
- **Small/Scalable** (Klein): Möglichst einheitlich kleine Anforderungen formulieren
- **Testable** (Testbar): Akzeptanzkriterien müssen formuliert sein

MoSCoW-Priorisierung

- Unterteilung der Anforderungen in 4 Gruppen
 - Must have
 - Should have
 - Could have
 - Won't have
- Kriterien zur Einteilung sowie die Benennung und Bedeutung der Kategorien müssen selbst festgelegt werden

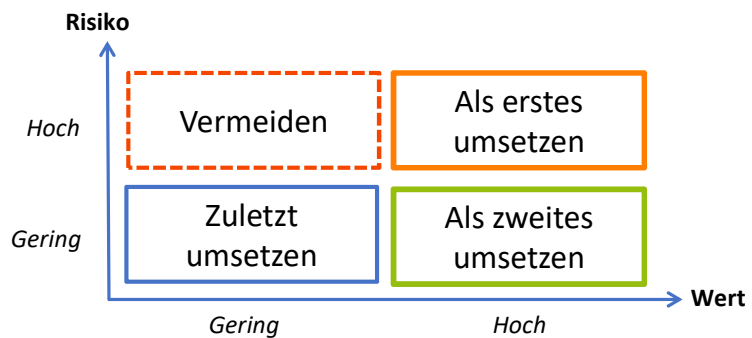


Die MoSCoW-Priorisierung hilft darzustellen, welche Anforderungen in einer aktuellen „Version“ umgesetzt werden – und welche aktuell nicht (mehr) im Scope sind

Priorisierung

- Priorisierung erfolgt i.d.R. nach Kundennutzen, Aufwand und Risiko
- Hoch priorisierte Anforderungen werden zuerst umgesetzt
 - Werden so definitiv Bestandteil des Produkts
 - Können frühzeitig vom Product Owner und Kunden überprüft werden
 - Entwicklungserfolg hängt vor allem von diesen Anforderungen ab

- Wert-Risiko-Matrix nach Mike Cohn



Aufwandsschätzungen

- Scrum macht keine Vorgaben, welche Größe für Aufwandsschätzungen verwendet werden muss

- Etablierte Größe, um Komplexitäten (nicht Zeiten) zu schätzen:

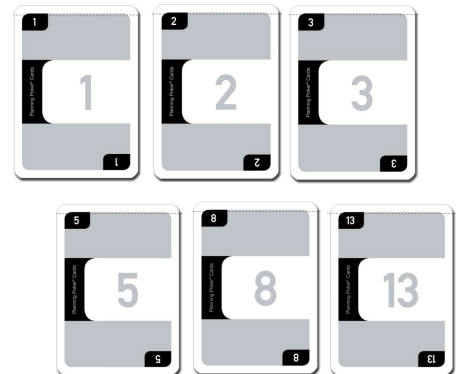
Story Points

- Verwendung einer Art Fibonacci-Reihe:

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

- Alternative Schätzgrößen

- Ideale Personentage (ebenfalls mit Fibonacci möglich)
- T-Shirt-Größen (XS, S, M, L, XL, XXL)

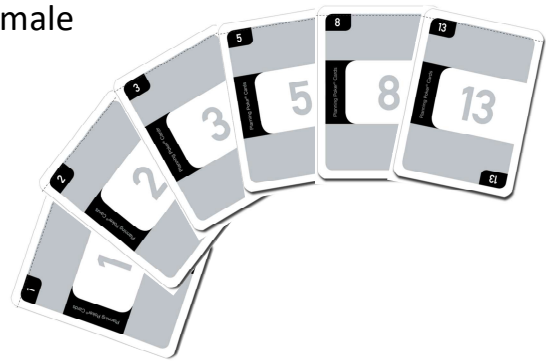


Story Points

Das Entwicklungsteam bewertet Referenzanforderungen, um ein gemeinsames Verständnis für die Story Points zu bekommen. Aufwandsabschätzungen sind immer Vergleiche zu Referenzanforderungen

Planning Poker

- Planning Poker ist eine beliebte Methode zur Aufwandsabschätzung
 - Anforderungen und User Stories werden gemeinsam vom Entwicklungsteam und dem Product Owner diskutiert
 - Das Team einigt sich auf die Verwendung einer Punktereihe (sehr kleiner Aufwand bis riesiger Aufwand)
 - Alle geben gleichzeitig eine Abschätzung der Anforderung ab
 - Bei unterschiedlichen Bewertungen werden der maximale und der minimale Wert diskutiert
- Vorteile von Planning Poker
 - Schnelle und transparente Teamentscheidung
 - Jeder muss unvoreingenommen eine eigene Schätzung abgeben



Sprints

- Überblick und Ablauf
- Sprint Planning, Daily SCRUM, Review & Retrospektive
- SCRUM Taskboard
- Impediments Backlog
- Fortschrittsmessung im Sprint

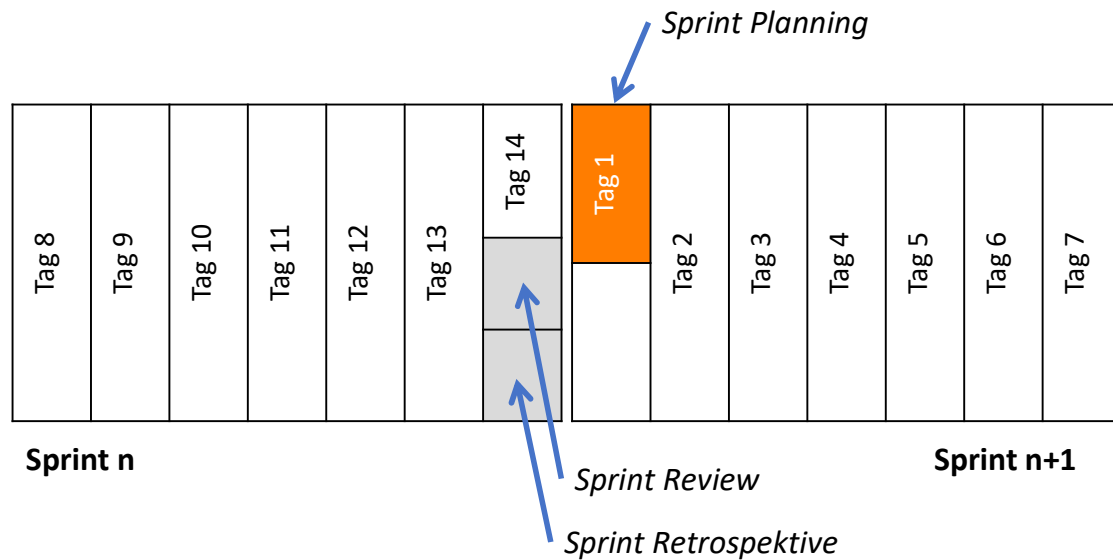
Überblick

- Sprintlänge: maximal 30 Tage
- werden in Wochen geplant (üblich sind 2 oder 3 Wochen)
- Ziel: Erweiterung des aktuellen Produktinkrements um lauffähige Funktionalität
- Ablauf
 - Sprint Planning: Ziele und Anforderungen für den Sprint
 - Daily SCRUM: Tägliche kurze Abstimmung im Team
 - Sprint Review: Abnahme der Anforderungen durch den Product Owner
 - Sprint Retrospektive: Kontinuierliche Verbesserung
- Es gibt keine Aktivitäten außerhalb eines Sprints



Zusammenhang zweier Sprints

- Zwei Sprints folgen ohne Pause direkt aufeinander



Empfehlenswert ist es, Sprints wochenweise zu planen, um über den gesamten Entwicklungsverlauf hinweg eine kontinuierliche und planbare Basis (wann finden welche Meetings statt) zu haben.

Eigenschaften

- **Ziel jedes Sprints: Erweiterung des Produktinkrements** (iterativ-inkrementelle Entwicklung)
 - Wurde vom Entwicklungsteam erfolgreich getestet und dokumentiert
 - Abnahme durch Product Owner nach definierten Akzeptanzkriterien
 - Ergebnisse eines Sprints sind Input für den nächsten
- **Schutz vor Veränderungen**
 - Keine Veränderung von Anforderungen, Zeit und Team, die das Sprintziel gefährden → Fixierung im Sprint Planning
 - Voraussetzung für Selbstorganisation des Teams
- **Absolute Termintreue** (timeboxing): der Sprint wird nicht verlängert und nicht verkürzt
 - Abbruch eines Sprints nur, wenn ein Ereignis den Sprint sinnlos macht



Sprint Planning

SPRINT-
PLANUNGSSITZUNG

- **Termin / Dauer:** Beginn des Sprints, 2h pro Woche Sprint
- **Teilnehmer:** PO, SM, EWT, ggf. weitere Fachexperten
- **Ablauf und Inhalte:**
 - **PO:** stellt gewünschtes Sprintziel und damit verbundene PBIs vor
 - **Scrum Team:** erarbeitet gemeinsames Verständnis des Sprintziels und der Anforderungen
 - **EWT:** wählt Anforderungen aus und schätzt diese final
 - **SM:** moderiert
- **Ergebnisse / Ziele:**
 - Commitment auf ein gemeinsames Sprintziel
 - Realistisches Sprint Backlog
 - Tasks für die nächsten Tage

Sprint-Ziel

Das Sprint-Ziel ist ein übergeordneter Zweck für den Sprint, der durch die Implementierung der Product-Backlog-Einträge erreicht werden kann. Es gibt dem Entwicklungsteam Orientierung, warum sie dieses Produktinkrement bauen. Das Sprint-Ziel wird während des Sprint Planning erarbeitet. Das Sprint-Ziel bietet dem Entwicklungsteam eine gewisse Flexibilität in Bezug auf die im Sprint zu implementierende Funktionalität. Die ausgewählten Product-Backlog-Einträge bilden eine zusammenhängende Funktionalität, die als Sprint-Ziel angesehen werden kann. Das Sprint-Ziel kann aber auch jedes andere verbindende Element sein, welches das Entwicklungsteam - statt in verschiedene Richtungen zu laufen - zur Zusammenarbeit motiviert.

Sprint Planning

SPRINT- PLANUNGSSITZUNG



- **Gemeinsames Verständnis** der ausgewählten Anforderungen
 - Diskussion der Anforderungen und der Akzeptanzkriterien
 - Wie wird nachgewiesen, dass die Anforderung umgesetzt ist?
- **Aufteilen des Sprint Plannings in zwei Teile:**
 - Teil 1: Erarbeitung des gemeinsamen Sprintziels („Was?“)
 - Teil 2: Aufteilung geplanter Tätigkeiten (Tasks) und Aufteilung untereinander („Wie?“)
- **Identifizieren der Arbeitsschritte (Tasks)**
 - Design, Implementierung, Integration, Tests und Dokumentation
 - Arbeitsweisen (z.B. Pair Programming) berücksichtigen
 - Detaillierungsgrad: präzise (oft ca. 1 Nettotag pro Task)
- **Ende der Planungssitzung**
 - wenn die Nettokapazität des Sprints erreicht ist (Sprint voll geplant)
 - wenn die anberaumte Zeit (Timebox) abgelaufen ist

Teamkapazität im nächsten Sprint

Bruttokapazität

- Verfügbare Stunden pro Arbeitstag pro Teammitglied
- z.B. 2 Wochen Sprint, 5 Vollzeit-Teammitglieder mit 8h-Arbeitstag
→ $10 \times 5 \times 8 = 400\text{h}$

Nettokapazität

Bruttokapazität abzüglich

- aller Scrum Ereignisse
- aller weiteren Aufwände und Fehlzeiten
(auch die, die nichts mit der Entwicklung zu tun haben, z.B. Betrieb, Abteilungssitzungen, etc.)

- Die Nettokapazität ist meist ca. 20 – 30% geringer als die Bruttokapazität
- Zur Korrelation von Kapazität (in Zeit) und Aufwand / Komplexität (in Story Points) wird die Velocity des EWT benötigt.

SPRINT-
PLANUNGSSITZUNG



Wenn sich die Kapazität eines Sprints nicht leicht vorhersagen lässt (geänderte Teamgröße, verschiedene Komponenten o.ä.), kann es hilfreich sein, die Teamkapazität im nächsten Sprint zu „berechnen“

Daily Scrum



- **Termin / Dauer:** täglich, 15 Minuten
- **Teilnehmer:** EWT; PO & SM optional und passiv
- **Ziel:**
 - Überprüfung des Fortschritts in Richtung Sprintziel
 - Optimierung der Leistung und Zusammenarbeit
- **Inhalte:**
 - „Welche Tasks habe ich seit dem letzten Mal abgeschlossen?“
 - „Woran plane ich bis zum nächsten Mal zu arbeiten?“
 - „Werde ich an der Ausführung meiner Aktivitäten behindert?“
- Typischerweise Aktualisierung des Taskboards

Das Entwicklungsteam oder einzelne Mitglieder treffen sich häufig direkt nach dem Daily Scrum für detailliertere Diskussionen, Anpassungen oder Umplanungen der Arbeit im Sprint.

Probleme und Hindernisse werden nicht während des Daily Scrum diskutiert.

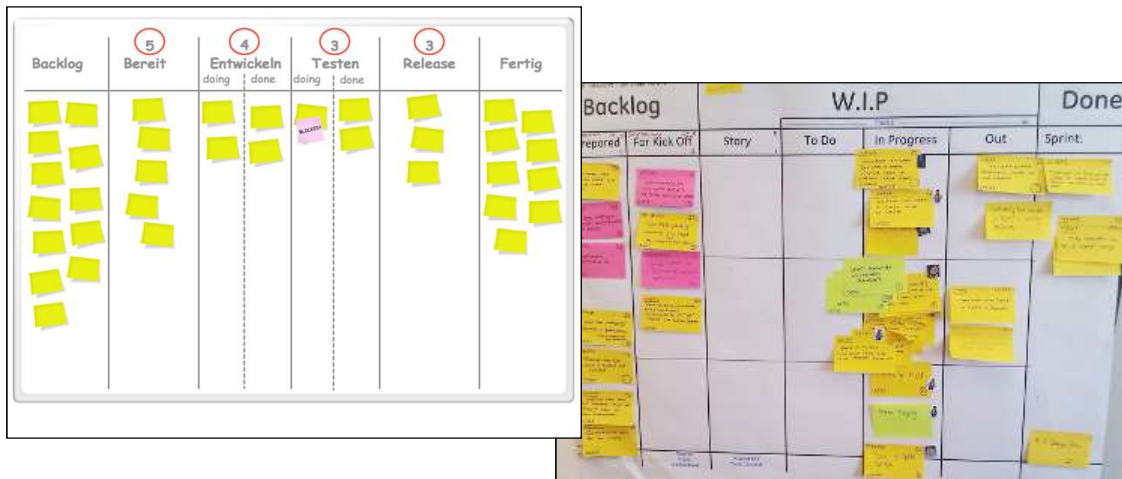
Es können verschiedene Techniken eingesetzt werden:

- Stand-up Meeting
- Speech Token
- Auszeit

Während der Scrum Master dafür sorgt, dass ein Daily Scrum stattfindet, ist das Entwicklungsteam für die Durchführung zuständig. Hierzu bringt der Scrum Master dem Entwicklungsteam bei, wie es die 15-minütige Time Box des Daily Scrums einhalten kann.

SCRUM Taskboard

- Zeitmanagementsystem zur Transparenz im Sprint
 - Übersicht aller Tasks (nach Kanban, in Kombination auch „Scrumban“)
 - Ständige Aktualisierung und Anpassung (mindestens einmal am Tag)



Vgl. <https://www.it-agile.de/wissen/einstieg-und-ueberblick/kanban/> und <https://kanbantool.com/scrumban>

Das SCRUM Taskboard soll ein „big picture“ (auch „information radiator“) des aktuellen Entwicklungsstands ermöglichen, so dass auf Reporting so weit wie möglich verzichtet werden kann. Interessierte Stakeholder können sich aktiv am Taskboard über den aktuellen Entwicklungsstand und Fortschritt informieren. Durch transparente Informationen wird Konversation begünstigt („Das Team hat nichts zu verbergen“). Eine optisch saubere und aktuelle Darstellung ist dabei unbedingt notwendig.

Sprint Review

SPRINT REVIEW



- **Termin / Dauer:** Ende des Sprints, 1h pro Woche Sprint
- **Teilnehmer:** PO, SM, EWT und Stakeholder
- **Ablauf und Inhalte:**
 - **PO:** reflektiert Sprintziel inkl. fertiger Anforderungen („Done“)
 - **EWT:** stellt den Verlauf des Sprints vor, führt die fertige Arbeit vor und beantwortet Fragen
 - **SM:** moderiert
 - **Alle Teilnehmer:** Gemeinsame Erarbeitung, was getan werden kann, um den Wert (Business Value) des Produktes / Projektes zu optimieren
- **Ergebnisse / Ziele:**
 - Feedback der Stakeholder
 - Überarbeitetes Product Backlog
 - Gemeinsames Verständnis des aktuellen Entwicklungsstandes

Atmosphäre: Offen, kollegial und ehrlich. Kritisch aber respektvoll

Sprint Retrospektive



- **Termin / Dauer:** Ende des Sprints, 45 min pro Woche Sprint
- **Teilnehmer:** PO, SM, EWT
- **Ablauf und Inhalte:**
 - **SM:** moderiert
 - Regeln erklären (z.B. ehrliche aber auch respektvolle Kommunikation)
 - Daten sammeln (positive und negative Vorkommnisse)
 - Ursachenanalyse der Daten
 - Maßnahmen beschließen (SMART)
 - Abschluss (Reflektion der Sitzung selbst)
- **Ergebnisse / Ziele:**
 - Diskutiertes Impediment Backlog
 - Verbesserungsplan für kommenden Sprint

Der Scrum Master bestärkt das Scrum-Team darin, seine Entwicklungsprozesse und -praktiken innerhalb des Scrum Prozessrahmenwerks zu verbessern, um im kommenden Sprint effektiver und befriedigender arbeiten zu können. In jeder Sprint Retrospektive erarbeitet das ScrumTeam Wege zur Verbesserung der Produktqualität durch die entsprechende Anpassung der Prozesse oder der Definition of Done, sofern diese Änderungen angemessen sind und nicht im Widerspruch mit Produkt- oder Unternehmensstandards stehen.

Impediments Backlog

- Angesprochene Hindernisse (Impediments) sollten ebenfalls transparent angesprochen und nachverfolgt werden (frühzeitige Eskalation)
- Da Hindernisse einen Workflow ähnlich wie Anforderungen haben, eignet sich auch hierfür ein eigenes Backlog in Taskboard-Darstellung

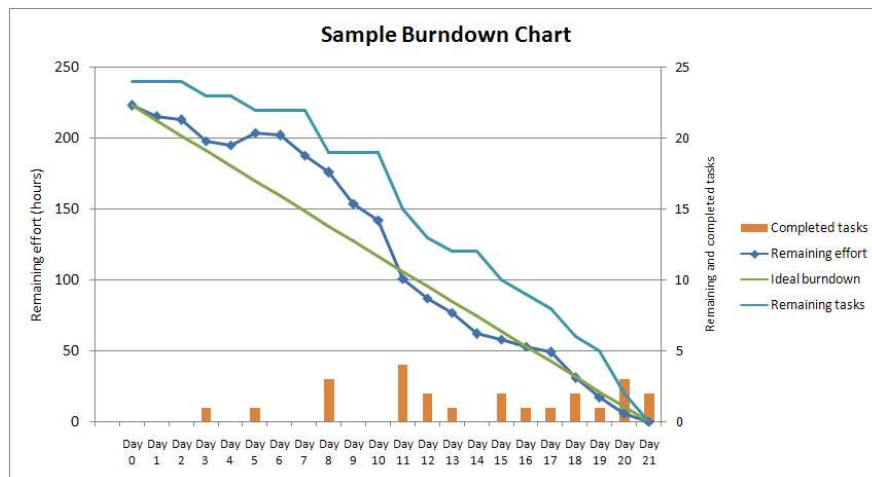
EIGNER	PROBLEM	WORKAROUND	LÖSUNG	ZIELDATUM	STATUS
	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	FERTIG
	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	~~~~~ ~~~~~ ~~~~~	IN ARBEIT

Vgl. <http://scrum-in-der-praxis.de/das-buch/illustrationen/3-8-globales-impediment-backlog/>

Fortschrittsmessung im Sprint

Sprint Burndown

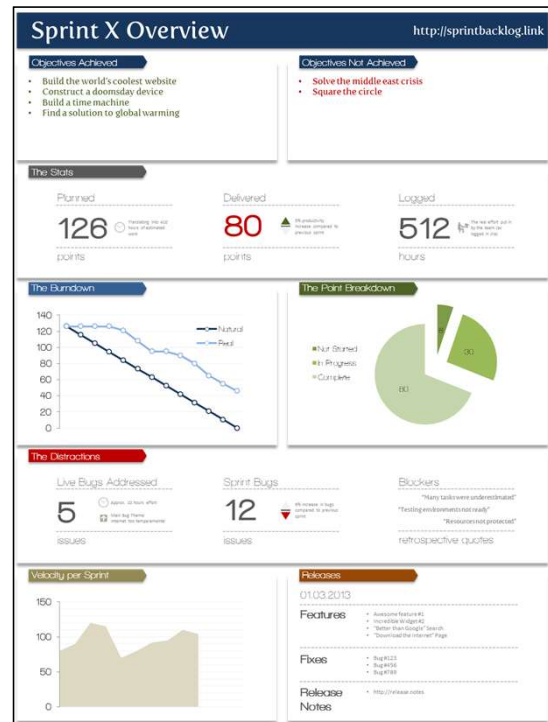
- Parallel zum Taskboard kann ein Sprint Burndown Chart gepflegt werden (Darstellung für Rückblick und Zielabschätzung geeignet)



Vgl. <https://de.wikipedia.org/wiki/Burn-Down-Chart>

End-of-Sprint Report

- Zusammenfassung aller Rahmendaten eines Sprints in einem Report
 - Geplante und abgenommene Anforderungen
 - Sprint Burndown Chart
 - Velocity (siehe später)
 - Impediments



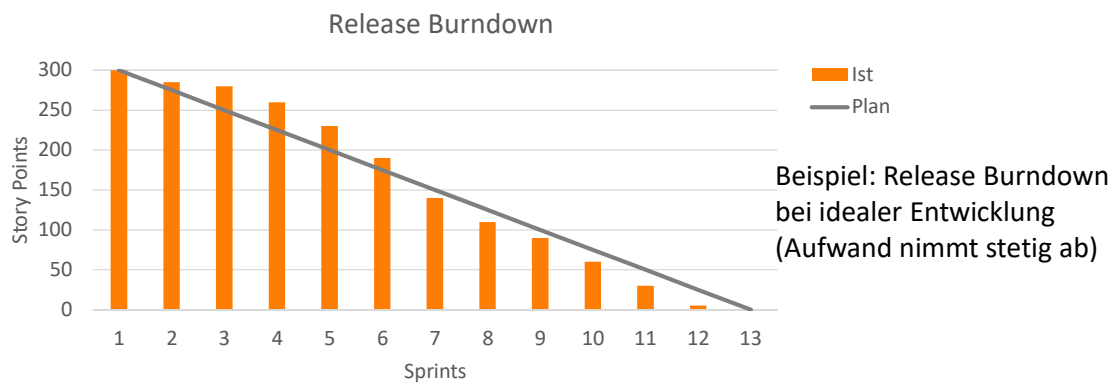
Vgl. <https://things-i-do.co.uk/2013/03/06/reporting-end-of-sprint-stats/>

Fortschrittsmessung (Metriken)

- Release Burndown
- Velocity (Entwicklungsgeschwindigkeit)
- Erfolgsrate

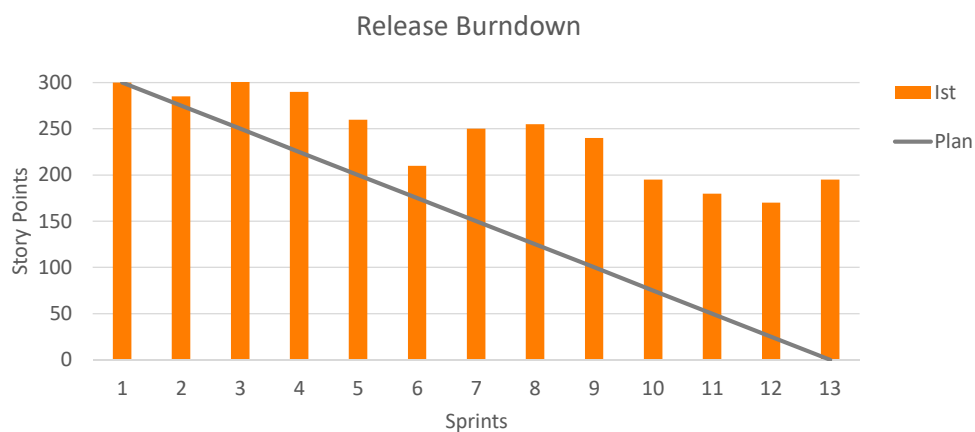
Release Burndown

- Darstellung des „Restaufwands“: Summe der restlichen Aufwände im Product Backlog
- Aktueller Entwicklungsfortschritt, Trend von einem zum nächsten Sprint als zukunftsweisende Messgröße (leading indicator)
- Vergleich realer und geplanter Fortschritt



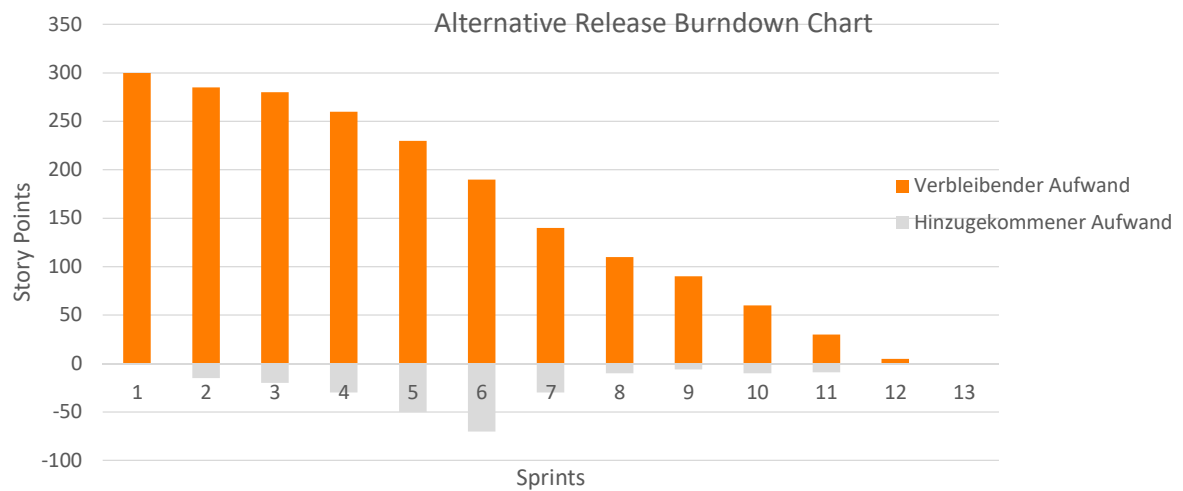
Release Burndown

- Beispiel: Fortlaufende Entwicklung bei ständig neuen und/oder sich ändernden Anforderungen
- Leichter Nachweis für Verzögerungen in der Entwicklung, bessere Abschätzung für zeitliche Verzögerung



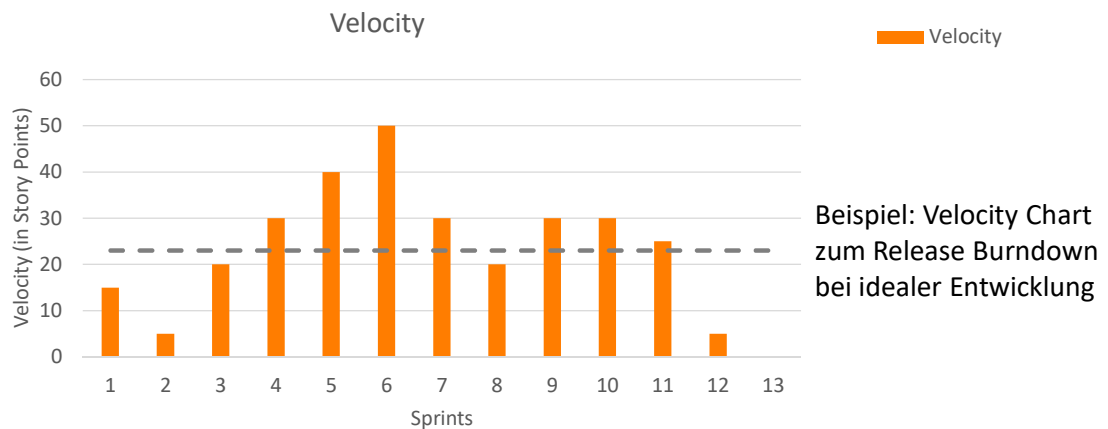
Alternative Release Burndown

- Differenzierte Darstellung von verbleibendem und hinzugekommenen Aufwand



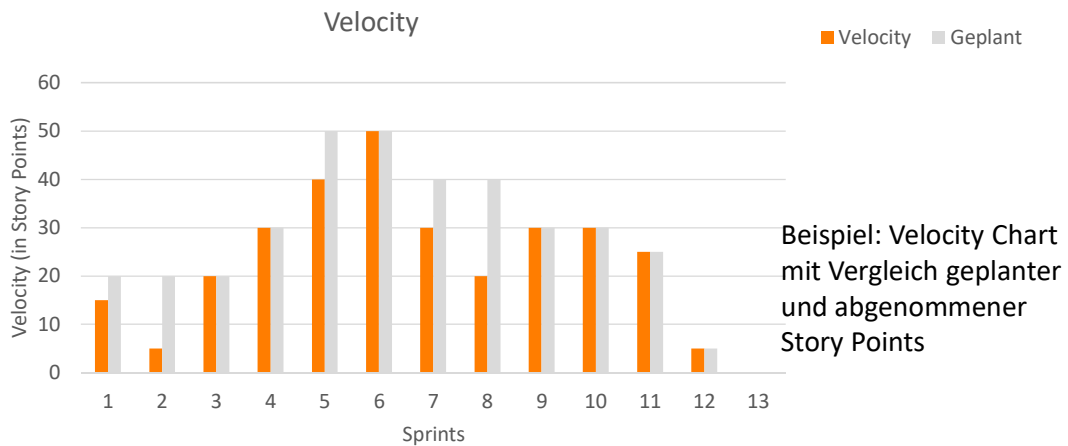
Velocity (Entwicklungsgeschwindigkeit)

- Die am Ende jedes Sprints vom Product Owner abgenommenen Arbeitsergebnisse mit ihren assoziierten Aufwänden (in Story Points)
 - Aufzeigen von Trends (z.B. Worst-case Releaseplan)
 - Nachweis, dass Hindernisse die Velocity negativ beeinflussen



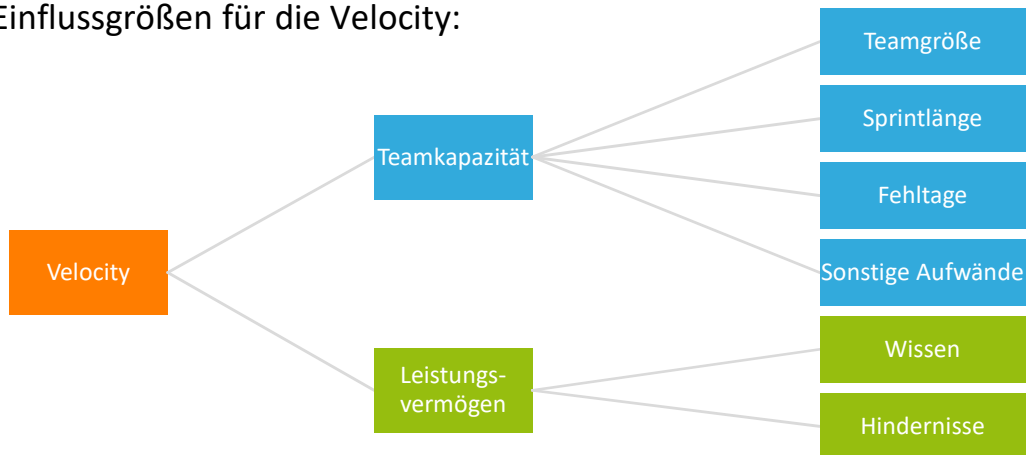
Velocity

- Vergleich von geplanten und tatsächlich erreichten Sprint-Zielen
- Erfolgsrate eines Sprints kann z.B. auch in Prozent oder mit Farbkodierung angegeben werden



Velocity

- Um vergleichbare und unverfälschte Werte zu erhalten, ist es wichtig, dass
 - nur teilweise fertiggestellte Ergebnisse nicht abgenommen werden
 - keine anteilmäßigen Punkte für teilweise Fertigstellung vergeben werden (Grad der Fertigstellung schwer abzuschätzen)
- Einflussgrößen für die Velocity:



Wenn Teamgröße oder Sprintlänge während der Entwicklung verändert werden, müssen natürlich die Metriken bzgl. der Velocity entsprechend angepasst werden, z.B. in Velocity pro Sprint-Tag oder Velocity pro Team-Mitglied, um vergleichbare Werte zu erhalten

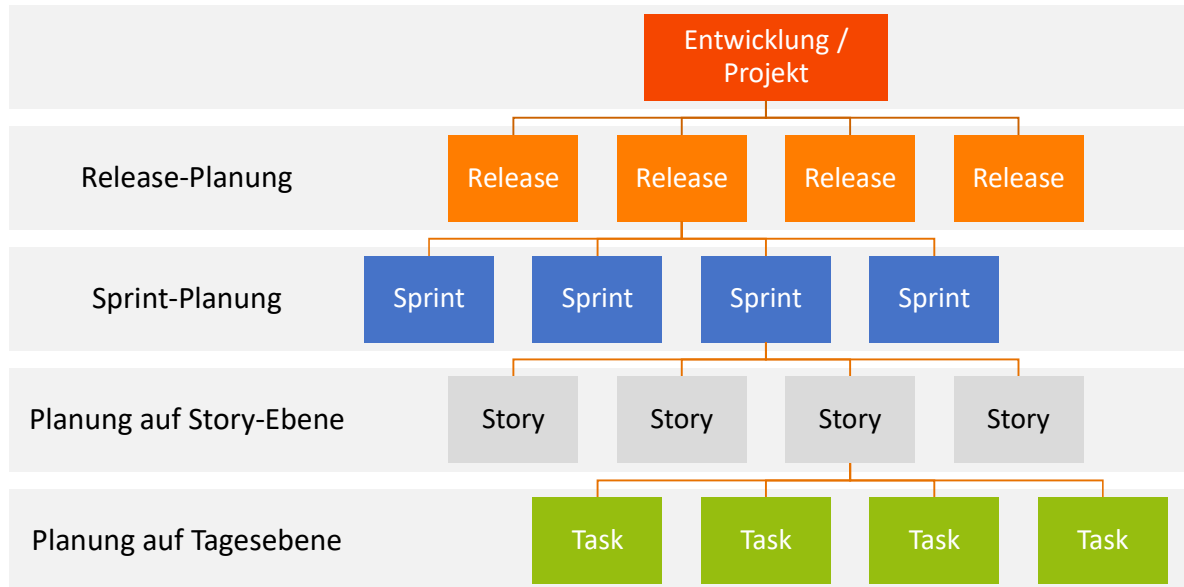
Weitere Metriken (Beispiele)

- Geplanter vs. erreichter Entwicklungsstand, z.B.
 - Prozent
 - Tage Abweichung
- Anzahl Hindernisse im Sprint
- Ergebnisse einzelner Teams
- Mitarbeiterzufriedenheit
- Fluktuation (vor allem in großen Projekten)
- Budget
- ...

Release Management in SCRUM

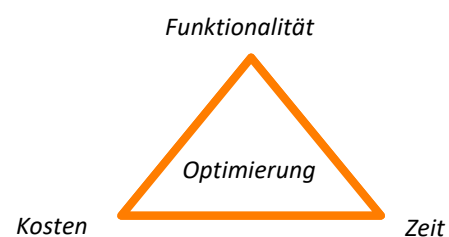
- Planungsebenen
- Ermittlung der Velocity
- Erstellung eines Releaseplans

Zusammenhänge der Planungsebenen



Überblick

- Releaseplanung ist in SCRUM optional und sollte eine ehrliche und transparente Vorhersage sein, um Zeit und Kosten abzuschätzen
- Für die Priorisierung und Optimierung der Arbeitsorganisation (Gemeinsames Verständnis einer sinnvollen Umsetzungsreihenfolge) sowie zur vorausschauenden Projektsteuerung kann ein Releaseplan hilfreich sein
- Der Releaseplan basiert auf
 - geschätztem Gesamtaufwand im Product Backlog
 - geschätzter Velocity



Schätzungen bzgl. des Gesamtaufwands im Product Backlog und der Velocity sind vor Entwicklungsbeginn mit entsprechenden Unsicherheiten verbunden. Hier unterscheidet sich SCRUM natürlich nicht von klassischem Projektmanagement. Allerdings kann in SCRUM relativ schnell nachvollzogen werden (Fortschritt verfolgen), wie hoch die Unsicherheit der anfänglichen Schätzungen ist.

Ermittlung der Velocity

- **Problem:**
Während der Entwicklung kann die Velocity einfach gemessen werden, für die Releaseplanung muss sie vor Beginn der Entwicklung ermittelt werden
- **Lösung 1:** Zurückgreifen auf historische Daten, falls dasselbe Team bereits ein ähnliches Produkt entwickelt hat
- **Lösung 2:** Schätzen
 - Nachteil: Unsicherheit
 - Velocity schwankt (bei Entwicklungsbeginn oft deutlich langsamer)
 - Teamkapazität kann sich im Entwicklungsverlauf ändern

Da die Velocity zu Entwicklungsbeginn oft deutlich niedriger ist als im späteren Verlauf, muss unbedingt darauf geachtet werden, dass das „negative“ Ergebnis der ersten Sprints nicht fälschlicherweise zu einem Entwicklungsabbruch oder ähnlichen Konsequenzen führt. Der SCRUM Master kann dabei helfen, die Daten richtig zu interpretieren.

Erstellung eines Releaseplans

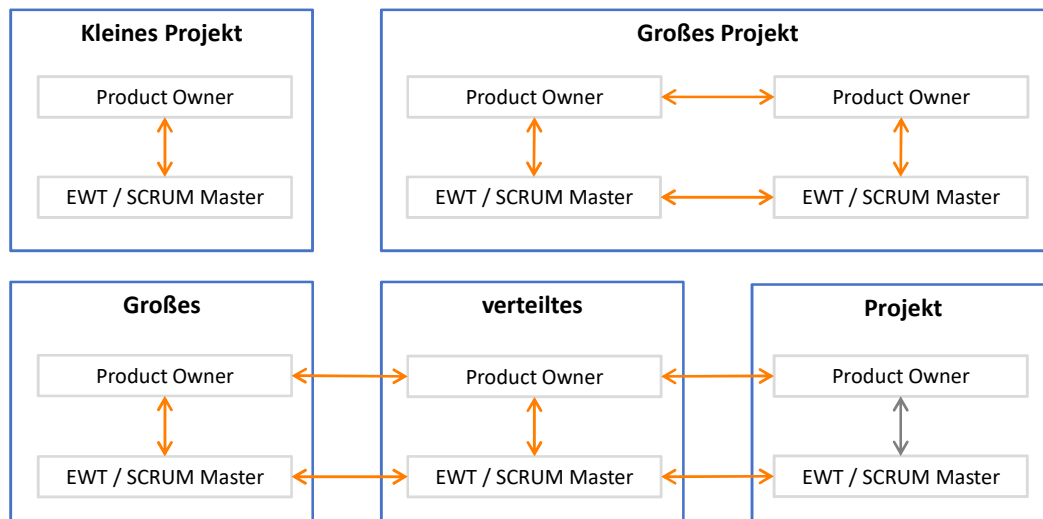
- **Entwicklungs-Zeitraum bestimmen**
 - Abgeschätzte Aufwände im Product Backlog
 - Ermittelte Velocity
 - Planungs-Workshops mit dem gesamten SCRUM Team
- **Festlegen der Reihenfolge** für Umsetzung der Anforderungen
 - Aufteilung in Features, Priorisierung
 - Zuweisung eines „Wunsch“-Ziels für jeden Sprint
- **Abstimmung von Release-Terminen**
 - An jedem Sprint-Ende, nach mehreren Sprints oder erst nach vollständigem Abschluss der Entwicklung
 - Der Releaseplan dient der Optimierung der Kundenzufriedenheit und sollte daher glaubwürdig sein

Große und verteilte Projekte

- Grundlagen und Wachstum in der Entwicklung
- Product Owner Team
- Feature- und Komponententeams
- SCRUM of SCRUMS
- Large-Scale SCRUM (LeSS)

Grundlagen

- Großes Scrum Projekt: wenn mehr als ein Entwicklungsteam beteiligt ist
- Verteiltes Scrum Projekt: wenn an mehr als an einem Ort gearbeitet wird



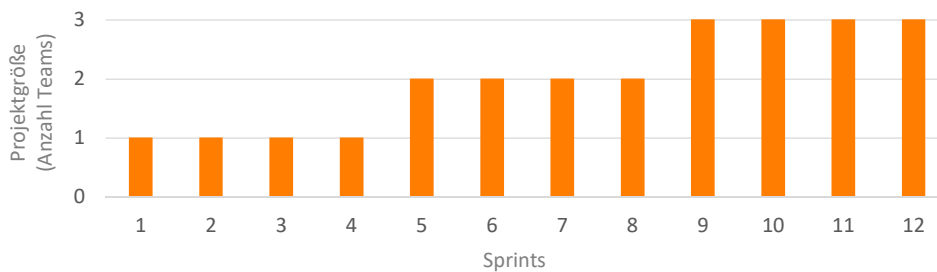
Da die meisten Grundlagen agiler Methoden wie Transparenz oder persönliche Kommunikation bereits bei mehreren Teams oder mehreren Standorten deutlich schwieriger zu erreichen sind, gelten bereits Projekte ab zwei Teams oder bei einer Verteilung auf zwei Standorte als „große“ bzw. „verteilte“ Projekte.

Herausforderungen

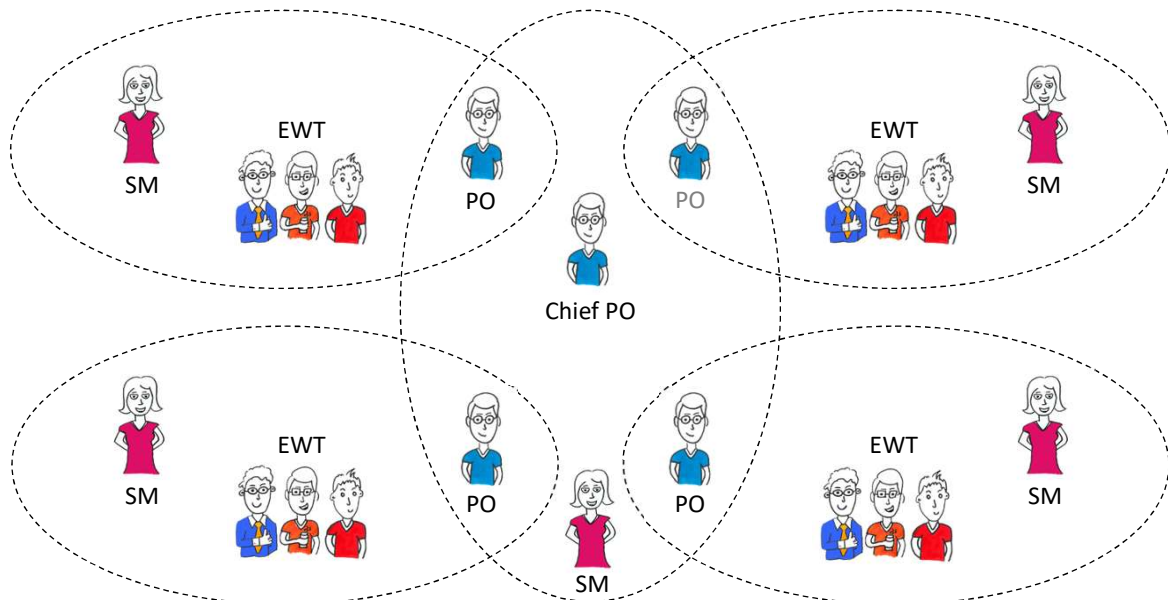
- Bevollmächtigung der einzelnen Entwicklungsteams ist Grundvoraussetzung
- Selbstorganisation der einzelnen Entwicklungsteams:
 - Entwicklungsteams können autonom handeln
 - Entwicklungsteams haben einen eigenen SCRUM Master und i.d.R. auch einen eigenen Product Owner
- Integration und Kommunikation schwierig
 - In großen Projekten müssen sich alle Mitglieder mit dem Projektziel identifizieren können
 - Gemeinsame Normen sind essentiell
 - Es empfiehlt sich, Projekte und die Entwicklungsteam-Anzahl organisch wachsen zu lassen

Grundlagen und Wachstum

- Vor der Skalierung müssen Grundlagen geschaffen werden
 - Etablierung der projektspezifischen Anwendung von SCRUM
 - Erstellung eines Produktkonzepts
 - Entscheidungen über Basisarchitektur, Standards und Normen
 - Bereitstellung von Test- und Entwicklungsumgebungen
- Projekte sollten langsam wachsen, ein bestehendes Entwicklungsteam aufbrechen ist meist besser als ein neues hinzuzufügen



Product Owner Team



Chief Product Owner

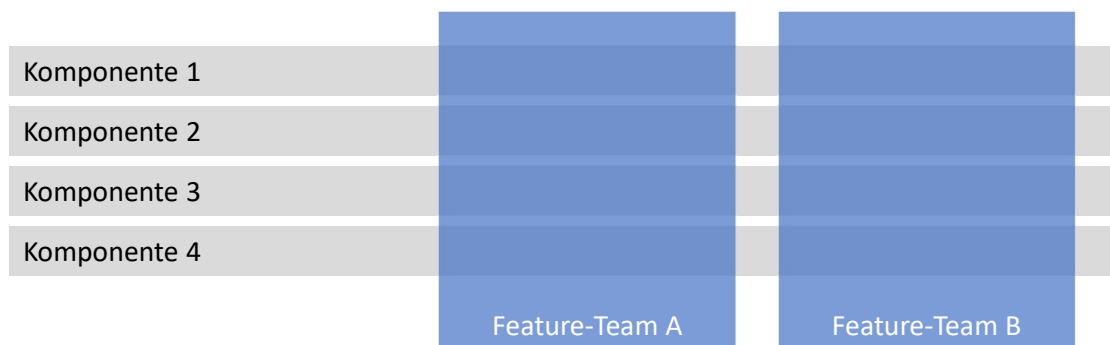
- Verantwortlich für das Gesamtprodukt
- Unterstützt durch ein Product Owner Team
- Kein direkter Zugriff auf die Entwicklungsteams

Product Owner Team als SCRUM Entwicklungsteam

- Verantwortlich für Beschreibung und Priorisierung der Anforderungen
- Chief Product Owner hat das letzte Wort
- Kann sich nach SCRUM-Regeln selbstorganisieren (mit eigenem SCRUM Master)

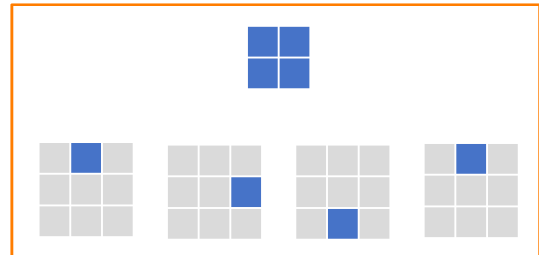
Feature- und Komponententeams

- Die Aufteilung in mehrere Teams kann nach technischen Komponenten oder analog zum Splitting der User Stories und passend zum multidisziplinären Team-Gedanken nach funktionalen Features erfolgen
 - Komponenten-Entwicklungsteams haben die Verantwortung für ein technisches Subsystem
 - Feature-Entwicklungsteams sind nach funktionalen Anforderungen multidisziplinär (Features) organisiert

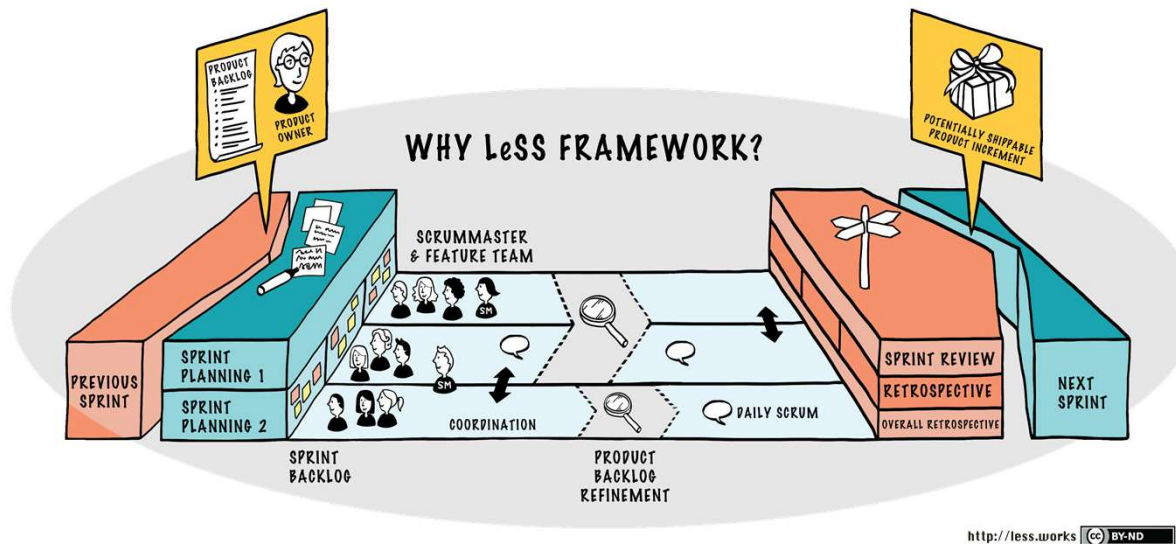


Multiteamkoordination „SCRUM of SCRUMS“

- Ziel
 - Unterstützung der projektweiten Selbstorganisation
 - Identifikation projektweiter Hindernisse
- Eigenschaften
 - Eine Art Daily SCRUM (oft wöchentlich)
 - Besteht aus Vertretern aller Entwicklungsteams
- Inhalt
 - „Was hat mein Team seit dem letzten Mal erledigt?“
 - „Woran plant mein Team bis zum nächsten Mal zu arbeiten?“
 - „Welche Hindernisse beeinträchtigen mein Team?“



Large-Scale SCRUM (LeSS)



Weitere Themen

- Vertragliche Aspekte mit SCRUM
- Toolunterstützung

Vertragliche Aspekte

- Werkvertrag vs. Dienstvertrag
 - Da das Produktinkrement häufig ein Werk darstellt, handelt es sich bei Entwicklungsverträgen i.d.R. um Werkverträge
 - Allerdings wird das zu entwickelnde Werk erst im Entwicklungsverlauf und nicht bereits zu Beginn ausreichend spezifiziert (widerspricht deutschem Vertragsrecht)
- Festpreisprojekte sind entsprechend kritisch (evtl. mit MoSCoW arbeiten)
 - Gemeinsame Schätzungen durchführen
 - Bereits erprobte Teams einsetzen
 - Kunde muss verstanden haben, wie SCRUM funktioniert

Typische Vertragsarten für Entwicklungsprojekte (c't 2017, Heft 23).

Kaufvertrag (§ 433 ff. BGB)

Kaufgegenstand ist eine Sache oder ein Recht (etwa eine Nutzungslizenz), nicht Arbeitsleistung oder sonstige Dienste. Der Kaufpreis ist mit Übergabe des Kaufgegenstands fällig, sofern nichts anderes vereinbart wurde. Der Verkäufer haftet nach 434 ff. BGB für Mängel der Kaufsache, muss aber die Möglichkeit der Nachbesserung bekommen. Das bedeutet, der Käufer muss ihm Gelegenheit geben, gerügte Mängel zu beheben.

Werkvertrag (§ 633 ff. BGB)

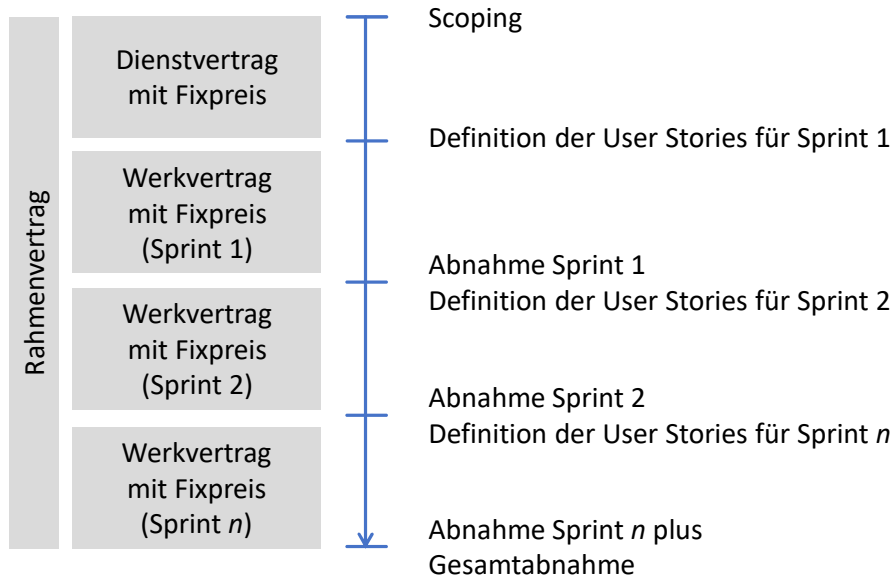
Geschuldet ist ein konkretes Ergebnis entsprechend der zuvor erfolgten Vereinbarung. Die Vergütung dafür ist mit der Abnahme fällig – also mit der Bestätigung, dass das Werk im Wesentlichen mangelfrei und funktionsfähig ist. Bekannte Mängel müssen vor Abnahme gerügt werden. Nachträglich auftretende Mängel lassen sich auch noch nach Abnahme geltend machen. Zur Fälligestellung von Abschlagszahlungen müssen die Teilleistungen abgenommen und dokumentiert werden (§63a BGB). Wie beim Kaufvertrag gilt, dass bei Mängeln Gelegenheit zur Nachbesserung zu geben ist.

Dienstvertrag (§ 612 ff. BGB)

Inhalt ist die Erbringung von (Arbeits-)leistung, nicht deren konkretes Ergebnis. Die Vergütung ist fällig, nachdem die Leistung erbracht wurde. Die Mängelrechte entsprechen denen des Kaufvertrags, eine Abnahme ist nicht nötig.

Vertragliche Aspekte

- Lösungsvorschlag, basierend auf der aktuellen Rechtslage*:
Kombination aus Dienst- und Werkverträgen



*) LG Wiesbaden, Urteil vom 30.11.2016 (Az. 11 O 10/15); OLG Frankfurt, Urteil vom 17.08.2017 (Az. 5 U 152/16), vgl. c't 2017, Heft 23

Toolunterstützung

- Die Entwicklung von SCRUM kann durch viele Tools unterstützt werden
 - Atlassian JIRA Software
 - Redmine
 - Teampulse
 - Scrumwise
 - ...

- Das verwendete Tool sollte so gut wie möglich ins Requirements Engineering, die Entwicklung und die Fortschrittsverfolgung integriert werden (Transparenz)
 - Source Code Integration
 - Modellierung (UML, Use Cases, etc.)
 - Validierungsmöglichkeiten (Akzeptanztests, etc.)



Munich Institute for
IT Service Management GmbH
Landaubogen 1
81373 München

Tel.: +49 89 - 55 27 55 70
office@mitsm.de
www.mitsm.de

Der Scrum Guide™

Der gültige Leitfaden für Scrum:
Die Spielregeln

November 2017



*Entwickelt und kontinuierlich verbessert von den Scrum-Erfindern: Ken Schwaber
und Jeff Sutherland*

Deutsche Ausgabe/German

Inhaltsverzeichnis

Zweck des Scrum Guides	3
Definition von Scrum	3
Anwendungen von Scrum	4
Scrum: Theorie	4
Scrum: Werte.....	5
Das Scrum-Team.....	6
Der Product Owner.....	6
Das Entwicklungsteam.....	7
Der Scrum Master.....	8
Scrum-Ereignisse	9
Der Sprint.....	9
Sprint Planning	10
Daily Scrum.....	12
Sprint Review.....	13
Sprint Retrospektive	14
Scrum Artefakte.....	14
Product Backlog.....	15
Sprint Backlog.....	16
Inkrement	17
Transparenz der Artefakte	17
Definition of “Done”	17
Schlussbemerkung.....	19
Danksagungen	19
Menschen.....	19
Historie	19
Übersetzung	20
Änderungshistorie der Übersetzung	20
Änderungen im Scrum Guide 2017 im Vergleich zur Version von 2016	20

Zweck des Scrum Guides

Scrum ist ein Rahmenwerk zur Entwicklung, Auslieferung und Erhaltung komplexer Produkte. Dieser Leitfaden beinhaltet die Definition von Scrum. Diese Definition besteht aus den Scrum Rollen, Ereignissen, Artefakten sowie den Regeln, die sie miteinander verbinden. Ken Schwaber und Jeff Sutherland haben Scrum entwickelt; der Scrum Guide wird von ihnen verfasst und veröffentlicht. Gemeinsam stehen sie hinter dem Scrum Guide.

Definition von Scrum

Scrum (n): Ein Rahmenwerk, innerhalb dessen Menschen komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt werden, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern.

Scrum ist:

- Leichtgewichtig
- Einfach zu verstehen
- Schwierig zu meistern

Scrum wird seit den frühen 1990er Jahren als Prozessrahmenwerk zum Management der Arbeit an komplexen Produkten verwendet. Scrum ist weder ein Prozess, noch eine Technik oder vollständige Methode. Vielmehr ist es ein Rahmenwerk, innerhalb dessen verschiedene Prozesse und Techniken zum Einsatz gebracht werden können. Scrum macht die relative Wirksamkeit Ihres Produktmanagements und Ihrer Arbeitstechniken sichtbar, damit Sie fortlaufend das Produkt, das Team und die Arbeitsumgebung verbessern können.

Das Scrum-Rahmenwerk besteht aus Scrum-Teams und den zu ihnen gehörenden Rollen, Ereignissen, Artefakten und Regeln. Jede Komponente innerhalb des Rahmenwerks dient einem bestimmten Zweck und ist unentbehrlich für den Einsatz von Scrum und dessen Erfolg.

Durch die Regeln von Scrum werden die Beziehungen und Wechselwirkungen zwischen den Rollen, Ereignissen und Artefakten bestimmt. Die Regeln von Scrum sind im Hauptteil dieses Dokumentes dargestellt.

Bestimmte Taktiken zur Nutzung von Scrum variieren und sind an anderer Stelle beschrieben.

Anwendungen von Scrum

Scrum wurde ursprünglich dazu entwickelt, um Produkte zu managen und zu entwickeln. Seit den frühen 1990er Jahren wird Scrum weltweit ausgiebig genutzt zur:

1. Erforschung und Identifizierung rentabler Märkte, Technologien und Produktfähigkeiten;
2. Entwicklung von Produkten und Erweiterungen;
3. Auslieferung von Produkten und Erweiterungen, regelmäßig und mehrmals täglich;
4. Entwicklung und Aufrechterhaltung von Cloud-Umgebungen (online, sicher, on-demand) und anderer Produktivumgebungen; sowie,
5. Erhaltung und Erneuerung von Produkten.

Scrum wird verwendet, um Software, Hardware, Embedded Software, Netzwerke von interagierenden Funktionen und autonome Fahrzeuge zu entwickeln. Scrum wird aber auch in Schulen, Regierungs- und Marketingprojekten genutzt, zur Verwaltung von Organisationen und der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen und als Gesellschaften verwenden.

Scrum bewährt sich täglich im Umgang mit Komplexität, da Technologie-, Markt- und Umweltkomplexitäten und deren Interaktionen rapide zugenommen haben.

Scrum hat sich bei iterativem und inkrementellem Wissenstransfer als besonders effektiv erwiesen. Scrum wird heute häufig für Produkte, Dienstleistungen und das Management der übergeordneten Organisation verwendet.

Der Kern von Scrum ist ein kleines Team von Menschen. Dieses Team ist sehr flexibel und anpassungsfähig. Diese Stärken wirken weiter zwischen mehreren und vielen Teams und sogar Netzwerken von Teams, die die Arbeit und Ergebnisse von Tausenden von Menschen entwickeln, ausliefern, betreiben und erhalten. Sie arbeiten über durchdachte Entwicklungsarchitekturen und Zielumgebungen für Auslieferungen zusammen.

Wenn die Wörter "entwickeln" und "Entwicklung" im Scrum Guide verwendet werden, beziehen sie sich auf komplexe Arbeit, wie z.B. die oben beschriebenen Typen.

Scrum: Theorie

Scrum basiert auf der Theorie empirischer Prozesssteuerung oder kurz "Empirie". Empirie bedeutet, dass Wissen aus Erfahrung gewonnen wird und Entscheidungen auf Basis des Bekannten getroffen werden. Scrum verfolgt einen iterativen, inkrementellen Ansatz, um Prognosesicherheit zu optimieren und Risiken zu kontrollieren.

Drei Säulen tragen jede empirische Prozesssteuerung: Transparenz, Überprüfung [Inspection] und Anpassung [Adaptation].

Transparenz

Die wesentlichen Aspekte des Prozesses müssen für diejenigen sichtbar sein, die für das Ergebnis verantwortlich sind. Transparenz erfordert, dass diese Aspekte nach einem gemeinsamen Standard definiert werden, damit die Betrachter ein gemeinsames Verständnis des Gesehenen teilen.

Dies umfasst beispielsweise:

- Eine gemeinsame Prozesssprache, die von allen Teilnehmern geteilt wird.
- Die Personen, die das Produktinkrement produzieren und inspizieren, müssen alle ein gemeinsames Verständnis von „fertig“ [„Done“] teilen.

Überprüfung

Scrum-Anwender müssen die Scrum-Artefakte und den Fortschritt regelmäßig in Bezug auf die Erreichung des Sprint-Ziels überprüfen, um unerwünschte Abweichungen zu erkennen. Ihre Untersuchungen sollten nicht so häufig erfolgen, dass sie die Arbeit behindern. Den größten Nutzen bringen Überprüfungen, wenn sie gewissenhaft durch kompetente Prüfer dort vorgenommen werden, wo die Arbeit verrichtet wird.

Anpassung

Wenn ein Prüfer feststellt, dass Aspekte des Prozesses von den akzeptablen Grenzwerten abweichen und dass das resultierende Produkt so nicht akzeptabel sein wird, müssen der Prozess oder das zu bearbeitende Material angepasst werden. Die Anpassung muss [dann] so schnell wie möglich vorgenommen werden, um weitere Abweichungen zu minimieren.

Scrum schreibt vier formale Ereignisse für Überprüfung und Anpassung vor. Diese werden im Abschnitt Scrum-Ereignisse beschrieben:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospektive

Scrum: Werte

Wenn die Werte Selbstverpflichtung, Mut, Fokus, Offenheit und Respekt durch das Scrum-Team verkörpert und gelebt werden, werden die Scrum-Säulen Transparenz, Überprüfung und Anpassung lebendig und bauen bei allen Beteiligten Vertrauen zueinander auf. Die Mitglieder des Scrum-Teams lernen und erforschen diese Werte, indem sie mit den Scrum-Ereignissen, Rollen und Artefakten arbeiten.

Der erfolgreiche Einsatz von Scrum beruht darauf, dass alle Beteiligten kompetenter bei der Erfüllung dieser fünf Werte werden. Sie verpflichten sich persönlich dazu, die Ziele des Scrum-Teams zu erreichen. Die Mitglieder des Scrum-Teams haben den Mut, das Richtige zu tun und an

schwierigen Problemen zu arbeiten. Jeder fokussiert sich auf die Arbeit im Sprint und die Ziele des Scrum-Teams. Das Scrum-Team und seine Stakeholder sind sich einig, offen mit allen Belangen ihrer Arbeit und den damit verbundenen Herausforderungen umzugehen. Mitglieder von Scrum-Teams respektieren sich gegenseitig als fähige, eigenverantwortliche Individuen

Das Scrum-Team

Ein Scrum-Team besteht aus dem Product Owner, dem Entwicklungsteam, sowie dem Scrum Master. Scrum-Teams sind selbstorganisierend und interdisziplinär. Selbstorganisierende Teams entscheiden selbst, wie sie ihre Arbeit am besten erledigen, anstatt dieses durch andere Personen außerhalb des Teams vorgegeben zu bekommen. Interdisziplinäre Teams verfügen über alle Kompetenzen, die erforderlich sind, um die Arbeit zu erledigen, ohne dabei von Personen außerhalb des Teams abhängig zu sein. Das Team-Modell in Scrum wurde konzipiert, um Flexibilität, Kreativität und Produktivität zu optimieren. Es hat sich herausgestellt, dass ein Scrum-Team für alle eingangs aufgeführten Anwendungsfälle und jegliche komplexe Arbeit in zunehmenden Maße effektiv ist.

Scrum-Teams liefern Produkte iterativ und inkrementell und maximieren somit die Gelegenheiten für Feedback. Die inkrementelle Auslieferung eines "fertigen" [Done] Produkts sorgt dafür, dass stets eine potentiell nützliche Version des Produkts zur Verfügung steht.

Der Product Owner

Der Product Owner ist dafür verantwortlich, den Wert des Produktes zu maximieren, das aus der Arbeit des Entwicklungsteams entsteht. Wie dies geschieht, kann je nach Organisation, Scrum-Team und Einzelpersonen stark variieren.

Der Product Owner ist die einzige Person, die für das Management des Product Backlogs verantwortlich ist. Das Product-Backlog-Management umfasst:

- Die Product-Backlog-Einträge klar zu formulieren;
- Die Einträge im Product Backlog so zu sortieren, dass Ziele und Missionen optimal erreicht werden können;
- Den Wert der Arbeit zu optimieren, die das Entwicklungsteam erledigt;
- Sicherzustellen, dass das Product Backlog sichtbar, transparent und für alle klar ist sowie zeigt, woran das Scrum-Team als nächstes arbeiten wird; und
- Sicherzustellen, dass das Entwicklungsteam die Product-Backlog-Einträge im erforderlichen Maße versteht.

Der Product Owner kann die oben genannten Arbeiten selbst durchführen oder sie durch das Entwicklungsteam erledigen lassen. Der Product Owner bleibt jedoch immer rechenschaftspflichtig.

Der Product Owner ist eine einzelne Person, kein Komitee. Er kann zwar die Wünsche eines Komitees im Product Backlog wiedergeben, aber diejenigen, die einen Eintrag des Product Backlogs in seiner Priorität verändern möchten, müssen sich an den Product Owner wenden.

Damit der Product Owner erfolgreich sein kann, muss die gesamte Organisation seine Entscheidungen respektieren. Die Entscheidungen des Product Owners sind in Inhalt und Reihenfolge des Product Backlogs sichtbar. Niemand darf das Entwicklungsteam zwingen, andere Anforderungen zu bearbeiten.

Das Entwicklungsteam

Das Entwicklungsteam besteht aus Profis, die am Ende eines jeden Sprints ein fertiges [„Done“] Inkrement übergeben, welches potenziell auslieferbar ist. Im Sprint Review muss ein fertiges Inkrement vorhanden sein. Nur Mitglieder der Entwicklungsteams erstellen das Produktinkrement.

Entwicklungsteams sind von der Organisation so strukturiert und befähigt, dass sie ihre eigene Arbeit selbst organisieren und managen. Die daraus resultierende Synergie optimiert die Gesamteffizienz und -Effektivität des Entwicklungsteams.

Entwicklungsteams weisen die folgenden Eigenschaften auf:

- Sie sind selbstorganisierend. Niemand (nicht einmal der Scrum Master) sagt dem Entwicklungsteam, wie es aus dem Product Backlog potentiell auslieferbare Funktionalität erzeugen soll.
- Entwicklungsteams sind interdisziplinär. Sie haben als Team alle Fähigkeiten, die notwendig sind, um ein Produktinkrement zu erstellen.
- Scrum kennt für Mitglieder des Entwicklungsteams keine Titel. Dies ist unabhängig von der Arbeit, die diese Personen erledigen.
- Scrum kennt keine weiteren Unterteilungen innerhalb des Entwicklungsteams, ungeachtet der verschiedenen Themenfelder, mit denen das Team sich befasst, also z.B. "Test", "Architektur", "Betrieb" oder "Analyse".
- Einzelne Mitglieder des Entwicklungsteams können zwar spezialisierte Fähigkeiten oder Spezialgebiete haben, aber die Rechenschaftspflicht obliegt dem Team als Ganzem.

Größe des Entwicklungsteams

Die optimale Größe des Entwicklungsteams ist klein genug, um flink zu bleiben und groß genug, um bedeutende Arbeit innerhalb eines Sprints erledigen zu können. Weniger als drei Mitglieder des Entwicklungsteams reduzieren die Interaktion und führen zu geringeren Produktivitätssteigerungen [als bei größeren Teams]. Kleinere Entwicklungsteams können eventuell kein potentiell auslieferbares Produktinkrement liefern, da sie möglicherweise nicht über alle benötigten Fähigkeiten verfügen. Mehr als neun Mitglieder erfordern zu viel Koordination. Große Entwicklungsteams erzeugen eine zu hohe Komplexität, als dass ein empirischer Prozess nützlich wäre. Product Owner und Scrum Master zählen nicht zu dieser Zahl dazu, sofern sie nicht ebenso die Arbeit aus dem Sprint Backlog erledigen.

Der Scrum Master

Der Scrum Master ist dafür verantwortlich, Scrum entsprechend des Scrum Guides zu fördern und zu unterstützen. Scrum Master tun dies, indem sie allen Beteiligten helfen, die Scrum-Theorie, Praktiken, Regeln und Werte zu verstehen.

Der Scrum Master ist ein „Servant Leader“ für das Scrum-Team. Der Scrum Master hilft denjenigen, die kein Teil des Scrum-Teams sind, zu verstehen, welche ihrer Interaktionen mit dem Team sich hilfreich auswirken und welche nicht. Der Scrum Master hilft dabei, die Zusammenarbeit so zu optimieren, dass der durch das Scrum-Team generierte Wert maximiert wird.

Der Dienst des Scrum Masters für den Product Owner

Der Scrum Master dient dem Product Owner auf verschiedene Arten, unter anderem durch das:

- Sicherstellen, dass Ziele, Umfang und Produktdomäne von allen im Scrum-Team so gut wie möglich verstanden werden;
- Vermitteln von Techniken für eine effektive Verwaltung des Product Backlogs;
- Vermitteln eines Verständnisses für die Notwendigkeit klarer, prägnanter Product Backlog Einträge im Scrum-Team;
- Schaffen eines Verständnisses für Produktplanung in einem empirischen Arbeitsumfeld;
- Sicherstellen, dass der Product Owner weiß, wie er das Product Backlog so anordnet, dass es den größten Wert erzeugt;
- Vermitteln des richtigen Verständnisses von Agilität und ihrer Anwendung;
- Unterstützen bei der Durchführung von Scrum Ereignissen bei Bedarf oder auf Anfrage.

Der Dienst des Scrum Masters für das Entwicklungsteam

Der Scrum Master dient dem Entwicklungsteam auf verschiedene Arten, unter anderem durch das

- Coachen des Entwicklungsteams hin zu Selbstorganisation und funktionsübergreifender Teamarbeit;
- Unterstützen des Entwicklungsteams bei der Schaffung hochwertiger Produkte;
- Beseitigen von Hindernissen, die das Entwicklungsteam aufhalten;
- Unterstützen bei der Durchführung von Scrum-Ereignissen bei Bedarf oder auf Anfrage;
- Coachen des Entwicklungsteams in Organisationen, in denen Scrum noch nicht vollständig angenommen und verstanden wird.

Der Dienst des Scrum Masters an der Organisation

Der Scrum Master dient der Organisation auf verschiedene Arten, unter anderem durch das

- Leiten und Coachen der Organisation bei der Einführung von Scrum;
- Planen von Scrum-Implementierungen innerhalb der Organisation;
- Unterstützen von Kollegen und Stakeholdern, Scrum und empirische Produktentwicklung zu verstehen und umzusetzen;

- Auslösen von Veränderungen zur Produktivitätssteigerung des Teams;
- Zusammenarbeiten mit anderen Scrum Mastern, um die Effektivität von Scrum-Implementierungen innerhalb der Organisation zu verbessern.

Scrum-Ereignisse

In Scrum werden vorgeschriebene Ereignisse verwendet, um eine Regelmäßigkeit herzustellen und die Notwendigkeit anderer, nicht in Scrum definierter, Besprechungen zu minimieren. Alle Ereignisse sind befristet [time boxed], so dass jedes Ereignis eine maximale Dauer hat. Die Dauer eines Sprints steht zu seinem Beginn fest und darf weder gekürzt noch verlängert werden. Die anderen Ereignisse dürfen beendet werden, sobald sie ihren Zweck erfüllt haben. Dies stellt sicher, dass nur so viel Zeit wie nötig aufgewendet und Verschwendung vermieden wird.

Mit Ausnahme des Sprints als Container für alle anderen Ereignisse ist jedes Scrum-Ereignis eine formale Gelegenheit zur Überprüfung und Anpassung. Diese Ereignisse sind genau dazu gedacht, an den kritischen Stellen Transparenz und Überprüfung zu ermöglichen. Das Weglassen irgendeines dieser Ereignisse führt zu verringerter Transparenz und ist eine verpasste Gelegenheit den gegenwärtigen Stand zu erfassen [Inspect] und darauf zu reagieren [Adapt].

Der Sprint

Das Herz von Scrum ist der Sprint, ein Zeitraum [Time Box] von maximal einem Monat, innerhalb dessen ein fertiges ["Done"], nutzbares und potenziell auslieferbares Produktinkrement hergestellt wird. Alle Sprints innerhalb eines Entwicklungsvorhabens sollten die gleiche Dauer haben. Der neue Sprint startet sofort nach Abschluss des vorherigen Sprints.

Ein Sprint beinhaltet und umfasst das Sprint Planning, die Daily Scrums, die Entwicklungsarbeit, das Sprint Review und die Sprint Retrospektive.

Während des Sprints:

- werden keine Änderungen vorgenommen, die das Sprint-Ziel gefährden,
- wird der Qualitätsanspruch nicht geschmälert, und
- kann der Anforderungsumfang zwischen Product Owner und Entwicklungsteam geklärt und neu ausgehandelt werden, wenn sich neue Erkenntnisse ergeben haben.

Jeder Sprint kann als ein Projekt mit einem Zeithorizont von maximal einem Monat gesehen werden. Wie mit einem Projekt will man mit einem Sprint etwas Bestimmtes erreichen. Jeder Sprint hat ein Ziel, was gebaut werden soll, einen Entwurf und einen flexiblen Plan, der die Umsetzung, die Arbeit und das resultierende Produktinkrement in die richtige Richtung lenkt.

Sprints sind auf einen Kalendermonat beschränkt. Wenn der Zeithorizont eines Sprints zu groß gewählt wird, kann sich die Definition des Ergebnisses ändern, die Komplexität ansteigen und das Risiko erhöhen. Sprints ermöglichen Vorhersagbarkeit, indem sie mindestens einmal im Monat Überprüfung und Anpassungen des Fortschritts zu einem bestimmten Sprint-Ziel sicherstellen. Sprints reduzieren dazu noch das Risiko auf die Kosten eines Monats.

©2017 Ken Schwaber and Jeff Sutherland. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

Einen Sprint abbrechen

Ein Sprint kann vorzeitig, d.h. vor dem Ablauf seiner Time Box, abgebrochen werden. Dazu ist nur der Product Owner berechtigt, auch wenn er oder sie den Abbruch auf Anraten der Stakeholder, des Entwicklungsteams oder des Scrum Masters vornimmt.

Ein Sprint würde dann abgebrochen werden, wenn das Sprint-Ziel obsolet wird. Das kann vorkommen, wenn das Unternehmen seine Zielrichtung wechselt, oder sich andere Markt- oder technologische Rahmenbedingungen ändern. In der Regel sollte ein Sprint dann abgebrochen werden, wenn die Fortführung unter den gegenwärtigen Umständen keinen Sinn mehr macht. Allerdings ist ein Abbruch bei der kurzen Dauer der Sprints selten sinnvoll.

Wenn ein Sprint abgebrochen wird, werden alle abgeschlossenen und "Done" Product-Backlog-Einträge begutachtet. Wenn ein Teil der Arbeit potenziell auslieferbar ist, wird sie vom Product Owner meistens abgenommen. Alle unvollständigen Product-Backlog-Einträge werden neu geschätzt und wieder in das Product Backlog aufgenommen. Die bislang daran geleistete Arbeit verliert schnell an Wert, daher müssen diese Einträge häufiger neu geschätzt werden.

Sprint-Abbrüche verbrauchen Ressourcen, da sich alle Teammitglieder zum Starten eines neuen Sprints in einem Sprint Planning umorganisieren. Sprint-Abbrüche sind zudem oft schmerzhaft für das Scrum-Team; sie sind eher unüblich.

Sprint Planning

Im Sprint Planning wird die Arbeit für den kommenden Sprint geplant. Dieser Plan entsteht durch die gemeinschaftliche Arbeit des gesamten Scrum-Teams.

Das Sprint Planning ist für einen einmonatigen Sprint auf maximal 8 Stunden befristet [Time Box]. Bei kürzeren Sprints wendet man normalerweise weniger Zeit auf. Der Scrum Master sorgt dafür, dass das Ereignis stattfindet und die Teilnehmer dessen Zweck verstehen. Er bringt dem Scrum-Team bei, das Ereignis innerhalb der Frist erfolgreich abzuschließen.

Das Sprint Planning beantwortet die folgenden Fragen:

- Was ist in dem Produktinkrement des kommenden Sprints enthalten?
- Wie wird die für die Lieferung des Produktinkrements erforderliche Arbeit erledigt?

Punkt 1: Was kann in diesem Sprint fertiggestellt werden?

Das Entwicklungsteam erstellt eine Prognose über die Funktionalität, die im Sprint entwickelt werden soll. Der Product Owner beschreibt das Ziel, das mit dem Sprint erreicht werden soll, und die Product-Backlog-Einträge, welche - wenn sie in dem Sprint abgeschlossen werden - das Ziel erfüllen. Das ganze Scrum-Team erarbeitet gemeinsam ein Verständnis über die Arbeitsinhalte des Sprints.

Als Eingangsgrößen für das Meeting dienen das Product Backlog, das neueste Produktinkrement, die veranschlagte Kapazität des Entwicklungsteams im Sprint sowie die bisherige Leistung desselben. Ausschließlich das Entwicklungsteam bestimmt die Anzahl der

ausgewählten Product-Backlog-Einträge für den kommenden Sprint. Nur es selbst kann beurteilen, was im kommenden Sprint machbar ist.

Während des Sprint Plannings erarbeitet das Scrum-Team gemeinsam ein Sprint-Ziel. Das Sprint-Ziel bildet die Messlatte, die durch die Implementierung der Product-Backlog-Einträge im Sprint erreicht wird; es leitet das Entwicklungsteam in der Frage, warum es dieses Produktinkrement erstellt.

Punkt 2: Wie wird die ausgewählte Arbeit erledigt?

Nach der Vereinbarung des Sprint-Ziels und der Auswahl der Product-Backlog-Einträge für den Sprint entscheidet das Entwicklungsteam, wie es das Produktinkrement erstellen möchte, damit die Funktionalität in einen "Done"-Zustand gebracht werden kann. Als Sprint Backlog bezeichnet man die Auswahl der Product-Backlog-Einträge für den jeweiligen Sprint plus den Umsetzungsplan des Entwicklungsteams.

Das Entwicklungsteam beginnt normalerweise mit dem Entwurf des Systems und den Tätigkeiten, die notwendig sind, um ein funktionsfähiges Produktinkrement zu erstellen. Die Arbeiten können sich in der Größe oder dem geschätzten Aufwand unterscheiden. Auf jeden Fall sollte das Entwicklungsteam genug Arbeit planen, um zu prognostizieren, was es im kommenden Sprint schaffen zu können glaubt. Die für die ersten Sprint-Tage geplanten Arbeiten sind nach Abschluss des Meetings in kleinere Einheiten - oft von einem Tag oder weniger - zerlegt. Das Entwicklungsteam organisiert selbst, wie es die Arbeiten im Sprint Backlog angeht, sowohl im Sprint Planning als auch im Sprint selbst.

Der Product Owner kann dabei helfen, die ausgewählten Product-Backlog-Einträge zu klären und ggf. Kompromisse einzugehen. Wenn das Entwicklungsteam herausfindet, dass es sich zu viel oder zu wenig Arbeit vorgenommen hat, kann es die ausgewählten Product-Backlog-Einträge neu mit dem Product Owner aushandeln. Das Entwicklungsteam kann auch andere Teilnehmer zu dem Meeting einladen, um technische oder fachliche Unterstützung zu erhalten.

Am Ende des Sprint Plannings sollte das Entwicklungsteam in der Lage sein, Product Owner und Scrum Master zu schildern, wie es als selbstorganisiertes Team an der Erreichung des Sprint-Ziels und der Erstellung des gewünschten Produktinkrements arbeiten möchte.

Sprint-Ziel

Das Sprint-Ziel ist ein übergeordneter Zweck für den Sprint, der durch die Implementierung der Product-Backlog-Einträge erreicht werden kann. Es gibt dem Entwicklungsteam Orientierung, warum sie dieses Produktinkrement bauen. Das Sprint-Ziel wird während des Sprint Planning erarbeitet. Das Sprint-Ziel bietet dem Entwicklungsteam eine gewisse Flexibilität in Bezug auf die im Sprint zu implementierende Funktionalität. Die ausgewählten Product-Backlog-Einträge bilden eine zusammenhängende Funktionalität, die als Sprint-Ziel angesehen werden kann. Das Sprint-Ziel kann aber auch jedes andere verbindende Element sein, welches das Entwicklungsteam - statt in verschiedene Richtungen zu laufen - zur Zusammenarbeit motiviert.

Bei seiner Arbeit behält das Entwicklungsteam sein Sprint-Ziel vor Augen. Um dieses zu erreichen, implementiert es die entsprechende Funktionalität und Technologie. Falls es sich zeigt, dass der Arbeitsumfang von den ursprünglichen Erwartungen abweicht, handelt das Entwicklungsteam gemeinsam mit dem Product Owner eine Änderung des Sprint-Backlog-Umfangs für den laufenden Sprint aus.

Daily Scrum

Das Daily Scrum ist eine Time Box von 15 Minuten für das Entwicklungsteam. Das Daily Scrum findet an jedem Tag des Sprints statt. Das Entwicklungsteam plant dabei die Arbeit für die nächsten 24 Stunden. Es überprüft die Arbeitsergebnisse seit dem letzten Daily Scrum und prognostiziert die im Sprint bevorstehende Arbeit, um die Zusammenarbeit und Leistung des Teams zu optimieren. Um die Komplexität zu reduzieren, wird das Daily Scrum an jedem Tag zur gleichen Uhrzeit am gleichen Ort abgehalten.

Das Entwicklungsteam überprüft im Daily Scrum seinen Fortschritt in Richtung des Sprint-Ziels und den Trend bei der Abarbeitung der Sprint Backlog-Einträge. Das Daily Scrum erhöht die Wahrscheinlichkeit, dass das Entwicklungsteam sein Sprint-Ziel erreicht. Das Entwicklungsteam sollte Tag für Tag im Blick haben, wie es als selbstorganisiertes Team zusammenarbeiten möchte, um das Sprint-Ziel zu erreichen und das erwartete Inkrement zum Sprintende zu liefern.

Die Struktur des Ereignisses wird vom Entwicklungsteam festgelegt und kann auf unterschiedliche Weise durchgeführt werden, sofern die Erreichung des Sprint-Ziels im Fokus steht. Einige Entwicklungsteams verwenden Fragen, andere konzentrieren sich eher auf Diskussionen. Hier ist ein Beispiel, was verwendet werden könnte:

- Was habe ich gestern getan, das dem Entwicklungsteam geholfen hat, das Sprint-Ziel zu erreichen?
- Was werde ich heute erledigen, um dem Entwicklungsteam beim Erreichen des Sprint-Ziels zu helfen?
- Sehe ich irgendein Hindernis, das mich oder das Entwicklungsteam daran hindert, das Sprint-Ziel zu erreichen?

Das Entwicklungsteam oder einzelne Mitglieder treffen sich häufig direkt nach dem Daily Scrum für detailliertere Diskussionen, Anpassungen oder Umplanungen der Arbeit im Sprint.

Während der Scrum Master dafür sorgt, dass ein Daily Scrum stattfindet, ist das Entwicklungsteam für die Durchführung zuständig. Hierzu bringt der Scrum Master dem Entwicklungsteam bei, wie es die 15-minütige Time Box des Daily Scrums einhalten kann.

Das Daily Scrum ist ein internes Ereignis für das Entwicklungsteam. Falls andere Personen anwesend sind, stellt der Scrum Master sicher, dass sie das Meeting nicht stören.

Daily Scrums verbessern die Kommunikation, machen andere Meetings überflüssig, identifizieren zu beseitigende Hindernisse, fokussieren und fördern die schnelle

Entscheidungsfindung und erhöhen den Wissensstand des Entwicklungsteams. Das Daily Scrum ist ein entscheidendes Meeting zur Überprüfung und Anpassung.

Sprint Review

Am Ende eines Sprints wird ein Sprint Review abgehalten, um das Produktinkrement zu überprüfen und das Product Backlog bei Bedarf anzupassen. Während des Sprint Reviews beschäftigen sich das Scrum-Team und die Stakeholder gemeinsam mit den Ergebnissen des Sprints. Zusammen mit eventuellen Änderungen am Product Backlog während des Sprints bieten diese die Basis für die gemeinsame Arbeit an möglichen neuen, den Wert des Produkts steigernden Punkten. Beim Sprint Review handelt es sich um ein informelles Meeting, keinen Statusreport. Die Vorführung des Inkrements ist als Anregung für Feedback und die Basis für die Zusammenarbeit gedacht.

Für einen einmonatigen Sprint wird für dieses Meeting eine Obergrenze [Time Box] von vier Stunden angesetzt. Für kürzere Sprints wird in der Regel ein kürzerer Zeitrahmen veranschlagt. Der Scrum Master stellt sicher, dass das Event stattfindet und die Teilnehmer seinen Zweck verstehen. Der Scrum Master bringt allen Beteiligten bei, das Event innerhalb der Frist [Time Box] durchzuführen.

Das Sprint Review beinhaltet die folgenden Elemente:

- Die Teilnehmer, bestehend aus dem Scrum-Team und wichtigen Stakeholdern, die vom Product Owner eingeladen werden.
- Der Product Owner erklärt, welche Product-Backlog-Einträge fertig ["Done"] sind, und welche nicht.
- Das Entwicklungsteam stellt dar, was während des Sprints gut lief, welche Probleme aufgetaucht sind, und wie es diese Probleme gelöst hat.
- Das Entwicklungsteam führt die fertige Arbeit vor, und beantwortet Fragen zu dem Inkrement.
- Der Product Owner stellt den aktuellen Stand des Product Backlogs dar. Er gibt bei Bedarf eine aktualisierte Vorhersage von wahrscheinlichen Ziel- und Lieferterminen auf der Basis des Entwicklungsfortschritts.
- Alle Teilnehmer erarbeiten gemeinsam, was als nächstes zu tun ist, so dass das Sprint Review wertvollen Input für die kommenden Sprint Plannings liefert.
- Es erfolgt eine Begutachtung, ob sich durch die Marktsituation oder den möglichen Produkteinsatz neue Erkenntnisse über die wertvollsten nächsten Schritte ergeben haben.
- Anschließend werden Zeitplan, Budget, die potentiellen Eigenschaften sowie die Markterwartungen für das nächste zu erwartende Produkt-Release überprüft.

Das Ergebnis des Sprint Reviews ist ein überarbeitetes Product Backlog, das die möglichen Product-Backlog-Einträge für den kommenden Sprint enthält. Das Product Backlog kann auch umfassend umgearbeitet werden, um neue Chancen zu nutzen.

Sprint Retrospektive

Die Sprint Retrospektive bietet dem Scrum-Team die Gelegenheit, sich selbst zu überprüfen und einen Verbesserungsplan für den kommenden Sprint zu erstellen.

Sie findet zwischen dem Sprint Review und dem nächsten Sprint Planning statt. Für einen einmonatigen Sprint wird hierfür eine Obergrenze von drei Stunden angesetzt. Bei kürzeren Sprints ist das Meeting in der Regel kürzer. Der Scrum Master sorgt dafür, dass das Meeting stattfindet und alle Teilnehmer dessen Zweck verstehen.

Der Scrum Master sorgt dafür, dass das Meeting konstruktiv und produktiv ist. Der Scrum Master lehrt alle, die Time Box einzuhalten. Aufgrund seiner Verantwortung für den Scrum-Prozess nimmt der Scrum Master als gleichberechtigtes Mitglied an der Sprint Retrospektive teil.

Die Sprint Retrospektive wird durchgeführt, um

- zu überprüfen wie der vergangene Sprint in Bezug auf die beteiligten Personen, Beziehungen, Prozesse und Werkzeuge verlief;
- die wichtigsten gut gelaufenen Elemente und mögliche Verbesserungen zu identifizieren und in eine Reihenfolge zu bringen; und
- einen Plan für die Umsetzung von Verbesserungen der Arbeitsweise des Scrum-Teams zu erstellen.

Der Scrum Master bestärkt das Scrum-Team darin, seine Entwicklungsprozesse und -praktiken innerhalb des Scrum Prozessrahmenwerks zu verbessern, um im kommenden Sprint effektiver und befriedigender arbeiten zu können. In jeder Sprint Retrospektive erarbeitet das Scrum-Team Wege zur Verbesserung der Produktqualität durch die entsprechende Anpassung der Prozesse oder der Definition of Done, sofern diese Änderungen angemessen sind und nicht im Widerspruch mit Produkt- oder Unternehmensstandards stehen.

Zum Ende der Sprint Retrospektive sollte das Scrum-Team Verbesserungen für den kommenden Sprint identifiziert haben. Die Umsetzung dieser Verbesserungen im nächsten Sprint ist die Anpassungsleistung zur Selbstüberprüfung des Scrum-Teams. Auch wenn jederzeit Verbesserungen eingeführt werden können, bietet die Sprint Retrospektive eine formelle Gelegenheit, sich auf die Überprüfung und Anpassung zu fokussieren.

Scrum Artefakte

Die Artefakte von Scrum repräsentieren Arbeit oder Wert, um Transparenz sowie Möglichkeiten zur Überprüfung und Anpassung zu schaffen. Die in Scrum definierten Artefakte wurden speziell so entworfen, dass sie die Transparenz der wesentlichen Informationen maximieren, um für alle ein gleiches Verständnis über das Artefakt zu schaffen.

Product Backlog

Das Product Backlog ist eine geordnete Liste von allem, von dem bekannt ist, dass es im Produkt enthalten sein soll. Es dient als einzige Anforderungsquelle für alle Änderungen am Produkt. Der Product Owner ist für das Product Backlog, seine Inhalte, den Zugriff darauf und die Reihenfolge der Einträge verantwortlich.

Ein Product Backlog ist niemals vollständig. Während seiner ersten Entwicklungsschritte zeigt es die anfangs bekannten und am besten verstandenen Anforderungen auf. Das Product Backlog entwickelt sich mit dem Produkt und dessen Einsatz weiter. Es ist dynamisch; es passt sich konstant an, um für das Produkt klar herauszustellen, was es braucht, um seiner Aufgabe angemessen zu sein, im Wettbewerb zu bestehen und den erforderlichen Nutzen zu bieten. Sofern ein Produkt existiert, gibt es auch das dazugehörige Product Backlog.

Im Product Backlog werden alle Features, Funktionalitäten, Verbesserungen und Fehlerbehebungen aufgelistet, die die Änderungen an dem Produkt in zukünftigen Releases ausmachen. Ein Product-Backlog-Eintrag enthält als Attribute eine Beschreibung, die Reihenfolge, die Schätzung und den Wert. Product-Backlog-Einträge enthalten oft Testbeschreibungen, die ihre Vollständigkeit nachweisen, wenn sie fertig [„Done“] sind.

Das Product Backlog entwickelt sich mit dem Einsatz eines Produktes, dessen Wertsteigerung sowie durch das Feedback des Marktes zu einer längeren, ausführlicheren Liste. Anforderungen werden nie aufhören, sich zu ändern. Daher ist das Product Backlog ein lebendes Artefakt. Änderungen an den Geschäftsanforderungen, Marktbedingungen oder der Technologie können Änderungen am Product Backlog nach sich ziehen.

Häufig arbeiten mehrere Scrum-Teams gemeinsam an einem Produkt. Dann wird ein einziges Product Backlog benutzt, um die anstehende Arbeit am Produkt zu beschreiben. In diesem Fall kann ein Gruppierungsattribut für die Product-Backlog-Einträge verwendet werden.

Als Verfeinerung [Refinement] des Product Backlogs wird der Vorgang angesehen, in dem Details zu Einträgen hinzugefügt, Schätzungen erstellt, oder die Reihenfolge der Einträge im Product Backlog bestimmt werden. Die Verfeinerung ist ein kontinuierlicher Prozess, in dem der Product Owner und das Entwicklungsteam gemeinsam die Product-Backlog-Einträge detaillieren. Bei der Verfeinerung des Product Backlogs werden die Einträge begutachtet und revidiert. Das Scrum-Team bestimmt, wann und wie diese Verfeinerungsarbeit erfolgt. Sie sollte normalerweise nicht mehr als 10% der Kapazität des Entwicklungsteams beanspruchen. Der Product Owner kann jedoch jederzeit die Einträge im Product Backlog aktualisieren oder aktualisieren lassen.

Höher eingeordnete Product-Backlog-Einträge sind generell klarer und weisen mehr Details auf als niedrigere. Präzisere Schätzungen entstehen auf der Basis von größerer Klarheit und Detailtiefe - je niedriger der Rang, desto weniger Details sind bekannt. Die Product-Backlog-Einträge, mit denen sich das Entwicklungsteam im kommenden Sprint beschäftigen soll, werden so weit verfeinert, dass jeder von ihnen innerhalb des Sprints fertiggestellt werden kann. Die

Product-Backlog-Einträge, für die das der Fall ist, werden als bereit ["Ready"] für die Auswahl durch das Entwicklungsteam in einem Sprint Planning angesehen. Ein Product-Backlog-Eintrag entwickelt diesen Transparenzgrad in der Regel durch die oben beschriebenen Verfeinerungs-Aktivitäten.

Das Entwicklungsteam ist für alle Schätzungen verantwortlich. Der Product Owner kann das Entwicklungsteam dahingehend beeinflussen, dass er ihm beim Verständnis der Einträge hilft oder Kompromisse eingeht. Die endgültige Schätzung erfolgt immer von denen, die auch die Arbeit erledigen werden.

Überwachung der Zielerreichung

Die verbleibende Arbeit zur Erreichung eines Ziels kann jederzeit aufsummiert werden. Der Product Owner vermerkt diese gesamte verbleibende Arbeit mindestens zu jedem Sprint Review. Er vergleicht diesen Betrag mit der verbleibenden Arbeit in früheren Sprint Reviews, um den Fortschritt der Arbeiten im Verhältnis zur restlichen Zeit zu begutachten. Diese Information wird allen Stakeholdern präsentiert.

Zur Fortschrittsprognose werden diverse Planungspraktiken eingesetzt, wie Burndown- oder Burnupdiagramme. Diese haben sich als nützlich erwiesen, allerdings schmälern sie nicht die Bedeutung des empirischen Vorgehens. In komplexen Umgebungen lassen sich zukünftige Ereignisse nicht vorherbestimmen. Nur was bereits geschehen ist, gibt Anhaltspunkte für die zukunftsgerichtete Entscheidungsfindung.

Sprint Backlog

Das Sprint Backlog ist die Menge der für den Sprint ausgewählten Product-Backlog-Einträge, ergänzt um einen Plan, um das Produktinkrement zu liefern und das Sprint-Ziel zu erreichen. Das Sprint Backlog ist eine Prognose des Entwicklungsteams darüber, welche Funktionalität im nächsten Inkrement enthalten sein wird, sowie über die erforderliche Arbeit, um diese Funktionalität in einem fertigen [„Done“] Inkrement zu liefern.

Das Sprint Backlog macht die gesamte Arbeit sichtbar, die das Entwicklungsteam für notwendig erachtet, um das Sprint-Ziel zu erreichen. Um kontinuierliche Verbesserungen zu gewährleisten, umfasst es mindestens eine wichtige Prozessverbesserung, die in der vorherigen Retrospektive identifiziert wurde.

Das Sprint Backlog ist ein ausreichend detaillierter Plan, um den Fortschritt innerhalb des Sprints im Daily Scrum erkennen zu können. Das Entwicklungsteam passt das Sprint Backlog während des Sprints an; das Sprint Backlog entwickelt sich so während des Sprints. Diese Entwicklung erfolgt, während das Entwicklungsteam den Plan abarbeitet und mehr über die noch benötigten Schritte zur Erreichung des Sprint-Ziels lernt.

Wenn weitere Arbeiten erforderlich sind, werden sie vom Entwicklungsteam zum Sprint Backlog hinzugefügt. Wenn eine Arbeit durchgeführt wird oder abgeschlossen wurde, wird die Schätzung der verbleibenden Arbeit aktualisiert. Wenn sich Bestandteile des Plans als unnötig erweisen, werden sie entfernt. Nur das Entwicklungsteam kann sein Sprint Backlog während des Sprints

ändern. Das Sprint Backlog ist ein hochgradig sichtbares Echtzeit-Abbild der Arbeit, die das Entwicklungsteam plant, während des Sprints zu erreichen. Es gehört einzig und allein dem Entwicklungsteam.

Überwachung des Sprint-Fortschritts

Zu jedem Zeitpunkt im Sprint kann die gesamte verbleibende Arbeit an den Sprint-Backlog-Einträgen aufsummiert werden. Das Entwicklungsteam verfolgt diese gesamte Restarbeit mindestens zu jedem Daily Scrum, um die Wahrscheinlichkeit, das Sprint-Ziel zu erreichen, sichtbar zu machen. Durch die Nachverfolgung der verbleibenden Arbeit während des Sprints kann das Entwicklungsteam seinen Fortschritt steuern.

Inkrement

Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Product-Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints. Am Ende eines Sprints muss das neue Inkrement fertig ["Done"] sein; das heißt es muss in einem verwendbaren Zustand sein und die Definition of Done des Teams erfüllen. Ein Inkrement ist ein Gegenstand inspizierbarer, fertiger „Done“-Arbeit, der die Empirie am Ende des Sprints unterstützt. Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels. Es muss auch dann im einsatzfähigen Zustand sein, wenn der Product Owner es aktuell noch gar nicht ausliefern will.

Transparenz der Artefakte

Scrum basiert auf Transparenz. Entscheidungen zur Wertoptimierung und Risikokontrolle werden auf der Basis des festgestellten Zustands der Artefakte vorgenommen. Diese Entscheidungen haben in dem Maß eine solide Basis, in dem die Transparenz der Artefakte umfassend ist. Bei einer unvollständigen Transparenz können Entscheidungen auf Sand gebaut sein, Wert kann gemindert und das Risiko erhöht werden.

Der Scrum Master muss mit dem Product Owner, dem Entwicklungsteam und anderen Beteiligten herausfinden, ob die Artefakte wirklich vollkommen transparent sind. Es gibt Methoden für den Umgang mit fehlender Transparenz; der Scrum Master muss jedem dabei helfen, die bestmöglichen Praktiken zu finden und anzuwenden, wenn vollständige Transparenz nicht gewährleistet werden kann. Ein Scrum Master spürt mangelnde Transparenz durch die Überprüfung der Artefakte, die Erkennung von Mustern, intensives Zuhören, und die Erkennung von Abweichungen zwischen den erwarteten und den tatsächlichen Resultaten auf.

Die Aufgabe des Scrum Masters ist es, mit dem Scrum-Team und der Organisation an der Verbesserung der Transparenz der Artefakte zu arbeiten. Diese Arbeit bedeutet zumeist Lernen, Überzeugen und Verändern. Transparenz fällt einem nicht in den Schoß, sie ist ein Weg, den es zu beschreiten gilt.

Definition of “Done”

Es müssen alle verstehen, was „Done“ bedeutet, sobald ein Product-Backlog-Eintrag oder ein Produkt- Inkrement als „Done“ bezeichnet wird. Obwohl sich dies erheblich von Scrum-Team zu

Scrum-Team unterscheiden kann, müssen alle Teammitglieder ein gemeinsames Verständnis davon haben wann Arbeit fertig ist, um Transparenz zu gewährleisten. Dies erfolgt durch die Definition of Done des Scrum-Teams und wird dazu verwendet festzustellen, wann die Arbeit an einem Produktinkrement fertig ist.

Die gleiche Definition leitet das Entwicklungsteam bei der Entscheidung, wie viele Product-Backlog-Einträge es während des Sprint Plannings selektieren kann. Der Zweck eines jeden Sprints ist es, Inkremente potentiell auslieferbarer Funktionalität zu liefern, die der aktuellen Definition of Done des Scrum-Teams entsprechen.

Entwicklungsteams liefern jeden Sprint ein Inkrement an Produktfunktionalität. Dieses Inkrement ist vollständig verwendbar, so dass der Product Owner sich jederzeit dazu entscheiden kann, es zu releasen. Wenn die Definition of Done für ein Inkrement Teil der Konventionen, Standards oder Richtlinien der Entwicklungsorganisation ist, müssen alle Scrum-Teams diese als Minimalziel befolgen.

Wenn "Done" für ein Inkrement nicht Teil der Konvention der Entwicklungsorganisation ist, muss das Entwicklungsteam des Scrum-Teams eine für das Produkt geeignete Definition of Done formulieren [*Anm. d. Übers.:* Das heißt nicht, dass Product Owner und Scrum Master nicht eingebunden werden]. Wenn es mehrere Scrum-Teams gibt, die am selben System- oder Produktrelease arbeiten, müssen alle Entwicklungsteams aller Scrum-Teams gemeinsam eine Definition of Done erstellen.

Jedes Inkrement ist additiv zu allen früheren Inkrementen und gründlich getestet, um sicherzustellen, dass alle Inkremente gemeinsam funktionieren.

Von gereiften Scrum-Teams wird erwartet, dass sie ihre jeweilige Definition of Done erweitern, um strengere Kriterien für eine höhere Qualität sicherzustellen. Neue Einträge in der Definition of Done können dazu führen, dass noch zu erledigende Arbeit in früheren "Done" Produktinkrementen aufgedeckt wird. Jedes einzelne Produkt oder System sollte eine Definition of Done haben, welche den Standard für jegliche daran durchgeführte Arbeit darstellt.

Schlussbemerkung

Scrum ist kostenlos und wird in Form dieses Guides angeboten. Die Rollen, Artefakte, Ereignisse und Regeln von Scrum sind unveränderlich. Es ist zwar möglich, nur Teile von Scrum einzusetzen - das Ergebnis ist dann aber nicht Scrum. Scrum existiert nur in seiner Gesamtheit und funktioniert sehr gut als Container für andere Techniken, Methoden und Praktiken.

Danksagungen

Menschen

Von den Tausenden Menschen, die zu Scrum beigetragen haben, sollten wir diejenigen besonders hervorheben, die für Scrum in seinen Anfängen von besonderer Bedeutung waren. Jeff Sutherland arbeitete mit Jeff McKenna sowie John Scumniotales, und Ken Schwaber arbeitete mit Mike Smith sowie Chris Martin - und alle arbeiteten zusammen. Viele weitere haben in den folgenden Jahren einen Beitrag geleistet. Ohne deren Hilfe wäre Scrum nicht so ausgefeilt, wie es heute ist.

Historie

Ken Schwaber und Jeff Sutherland haben Scrum gemeinsam zum ersten Mal auf der OOPSLA Konferenz 1995 präsentiert. Diese Präsentation dokumentierte im Kern die Erkenntnisse, die Ken und Jeff in den vorher vergangenen Jahren bei der Anwendung von Scrum gewonnen hatten.

Die Geschichte von Scrum wird an anderer Stelle beschrieben. Zur Würdigung der ersten Orte, an denen Scrum ausprobiert und verfeinert wurde, möchten wir hier Individual, Inc., Newpage, Fidelity Investments und IDX (heute GE Medical) hervorheben.

Der Scrum Guide dokumentiert Scrum, so wie es seit über 20 Jahren von Jeff Sutherland und Ken Schwaber konzipiert und weiterentwickelt wird. Andere Quellen liefern Ihnen Modelle, Prozesse und Einsichten, die das Rahmenwerk von Scrum ergänzen. Damit lassen sich die Produktivität, der Wert, die Kreativität und der Stolz auf das Erreichte beständig erhöhen.

Übersetzung

Dieser Guide wurde von der englischen Originalversion, bereitgestellt von Ken Schwaber und Jeff Sutherland, übersetzt. Hierzu beigetragen haben:

2011: Dominik Maximini, Andreas Schliep, Ulf Schneider, Wolfgang Wiedenroth

2013: Jan Gretschuskin, Dominik Maximini, Pascal Naujoks, Sabrina Roos, Andreas Schliep, Wolfgang Wiedenroth

2016: Jan Gretschuskin, Dominik Maximini, Pascal Naujoks, Boris Steiner, Jürgen Halstenberg, Ralph Jocham, Wolf Dieter Moggert, Patrick Koglin, Harald Schlindwein

2017: Jan Gretschuskin, Ralph Jocham, Dominik Maximini, Stefan Mieth, Wolf Dieter Moggert, Pascal Naujoks, Urs Reupke, Andreas Schliep, Harald Schlindwein, Boris Steiner

Änderungshistorie der Übersetzung

- Version 1.0:** Komplette Neuübersetzung des Scrum Guides, basierend auf der Version von 2016.
- Version 1.1:** Klarstellung im Abschnitt „Das Scrum Team“: Interdisziplinäre Teams sind im Idealfall nicht von Personen außerhalb des Teams abhängig.
- Klarstellung im Abschnitt: „Anwendungen von Scrum“: Frequenz von Auslieferungen regelmäßig und mehrfach täglich.
- Version 1.2:** Klarstellung im Abschnitt: „Sprint Review“: Der Scrum Master agiert als Servant Leader und nicht nur als Organisator des Events.

Änderungen im Scrum Guide 2017 im Vergleich zur Version von 2016

1. Kapitel hinzugefügt: Anwendungen von Scrum

Scrum wurde ursprünglich dazu entwickelt, um Produkte zu managen und zu entwickeln. Seit den frühen 1990er Jahren wird Scrum weltweit ausgiebig genutzt zur:

1. Erforschung und Identifizierung rentabler Märkte, Technologien und Produktfähigkeiten;
2. Entwicklung von Produkten und Verbesserungen;

©2017 Ken Schwaber and Jeff Sutherland. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

3. Auslieferung von Produkten und Verbesserungen, auch vielfach pro Tag;
4. Entwicklung und Aufrechterhaltung von Cloud-Umgebungen (online, secure, on-demand) und anderer Produktivumgebungen; sowie,
5. Erhaltung und Erneuerung von Produkten.

Scrum wird verwendet, um Software, Hardware, Embedded Software, Netzwerke von interagierenden Funktionen und autonome Fahrzeuge zu entwickeln. Scrum wird aber auch in Schulen, Regierungs- und Marketingprojekten genutzt, zur Verwaltung von Organisationen und der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen und als Gesellschaften verwenden.

Scrum bewährt sich täglich im Umgang mit Komplexität, da Technologie-, Markt- und Umweltkomplexitäten und deren Interaktionen rapide zugenommen haben.

Scrum hat sich bei iterativem und inkrementellem Wissenstransfer als besonders effektiv erwiesen. Scrum wird heute häufig für Produkte, Dienstleistungen und das Management der übergeordneten Organisation verwendet.

Der Kern von Scrum ist ein kleines Team von Menschen. Dieses Team ist sehr flexibel und anpassungsfähig. Diese Stärken wirken weiter zwischen mehreren und vielen Teams und sogar Netzwerken von Teams, die die Arbeit und Ergebnisse von Tausenden von Menschen entwickeln, ausliefern, betreiben und erhalten. Sie arbeiten über durchdachte Entwicklungsarchitekturen und Zielumgebungen für Auslieferungen zusammen.

Wenn die Wörter "entwickeln" und "Entwicklung" im Scrum Guide verwendet werden, beziehen sie sich auf komplexe Arbeit, wie z.B. die oben beschriebenen Typen.

2. Die Formulierung im Kapitel „Der Scrum Master“ wurde geändert, um die Rolle klarer zu beschreiben. Der Text lautet jetzt:

Der Scrum Master ist dafür verantwortlich, Scrum entsprechend des Scrum Guides zu fördern und zu unterstützen. Scrum Master tun dies, indem sie allen Beteiligten helfen, die Scrum Theorie, Praktiken, Regeln und Werte zu verstehen.

Der Scrum Master ist ein „Servant Leader“ für das Scrum-Team. Der Scrum Master hilft denjenigen, die kein Teil des Scrum-Teams sind, zu verstehen, welche ihrer Interaktionen mit dem Team sich hilfreich auswirken und welche nicht. Der Scrum Master hilft dabei, die Zusammenarbeit so zu optimieren, dass der durch das Scrum-Team generierte Wert maximiert wird.

3. Einen Abschnitt im Kapitel „Der Dienst des Scrum Masters für den Product Owner“ hinzugefügt:

Sicherstellen, dass Ziele, Umfang und Produktdomäne von allen im Scrum-Team so gut wie möglich verstanden werden;

4. Aktualisierung des ersten Absatzes zum Daily Scrum:

Das Daily Scrum ist eine Time Box von 15 Minuten für das Entwicklungsteam. Das Daily Scrum findet an jedem Tag des Sprints statt. Das Entwicklungsteam plant dabei die Arbeit für die nächsten 24 Stunden. Es überprüft die Arbeitsergebnisse seit dem letzten Daily Scrum und prognostiziert die im Sprint bevorstehende Arbeit, um die Zusammenarbeit und Leistung des Teams zu optimieren. Um die Komplexität zu reduzieren, wird das Daily Scrum an jedem Tag zur gleichen Uhrzeit am gleichen Ort abgehalten.

5. Weitere Aktualisierung des Kapitels zum Daily Scrum, um die Ziele des Daily Scrums besser herauszustellen:

Die Struktur des Ereignisses wird vom Entwicklungsteam festgelegt und kann auf unterschiedliche Weise durchgeführt werden, sofern die Erreichung des Sprint-Ziels im Fokus steht. Einige Entwicklungsteams verwenden Fragen, andere konzentrieren sich eher auf Diskussionen. Hier ist ein Beispiel, was verwendet werden könnte:

- Was habe ich gestern getan, das dem Entwicklungsteam geholfen hat, das Sprint-Ziel zu erreichen?
- Was werde ich heute erledigen, um dem Entwicklungsteam beim Erreichen des Sprint-Ziels zu helfen?
- Sehe ich irgendein Hindernis, das mich oder das Entwicklungsteam daran hindert, das Sprint-Ziel zu erreichen?

6. Klarheit zu den Time Boxen hinzugefügt

Das Wort "maximal" hinzugefügt, damit es keine Fragen mehr dazu gibt, ob die Dauer einer Time Box voll ausgenutzt werden muss. Das stellt klar, dass es sich um Maximalzeiten handelt.

7. Ergänzung beim Sprint Backlog:

Um kontinuierliche Verbesserungen zu gewährleisten, umfasst es mindestens eine wichtige Prozessverbesserung, die in der vorherigen Retrospektive identifiziert wurde.

8. Klarheit beim Inkrement geschaffen:

Ein Inkrement ist ein Gegenstand inspizierbarer, fertiger [„Done“] Arbeit, der die Empirie am Ende des Sprints unterstützt. Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels.

Musterprüfung

SCRUM Foundation Examination

Einführung

Allgemeine Information

Die **Dauer** der Prüfung beträgt **45 Minuten**.

Die Prüfung erfolgt im **Single-Choice-Verfahren** und umfasst **30 Fragen**.

Die Prüfung gilt als **bestanden**, wenn Sie **20** oder mehr richtige Antworten haben.

Die **maximal erreichbare Punktzahl** ist **30**.

Bewertung

Pro Frage gibt es vier Antwortmöglichkeiten, von denen **eine Antwort** korrekt ist. Jede **richtig** beantwortete Frage gibt **einen Punkt**. Bei **falsch** beantworteten Fragen gibt es 0 Punkte (aber keinen Punktabzug). Eine Frage kann nicht mit weniger als 0 Punkte bewertet werden.

Papierbasierte Prüfungen

Alle Antworten sind auf dem beigegeführten Antwortbogen einzutragen. Nur diese werden bei der Auswertung berücksichtigt. Sämtliche Prüfungsunterlagen sind nach Abschluss dem Prüfer zurückzugeben.

Wir wünschen Ihnen viel Erfolg!

Copyright © TÜV SÜD Akademie GmbH

All rights reserved. No part of this publication may be published, reproduced, copied or stored in a data processing system or circulation in any form by print, photo print, microfilm or any other means without written permission by TÜV SÜD Akademie GmbH. All other company, product or service names may be trade marks or service marks of others and are the property of their respective owners.

FND Muster

1) Welche Aussage zum Scrum Master ist richtig?

- a) Die Rolle des Scrum Masters ist am ehesten mit der eines klassischen Projektmanagers vergleichbar
- b) Um Rivalitäten im Team vorzubeugen, sollte der Scrum Master vom Management ernannt werden
- c) Je weniger er zu tun hat, desto produktiver ist das Team
- d) Es ist von Vorteil, wenn der Scrum Master auch Personalverantwortung für die Teammitglieder hat

2) Für die Sprint-Planung ist es wichtig, dass Anforderungen gut formuliert werden. Hierbei sollten Anforderungen die INVEST-Eigenschaften erfüllen. Dabei gilt:

- a) Das "I" in INVEST steht für Innovation
- b) Das "S" in INVEST steht für Specific (zielgerichtet)
- c) Das "V" in INVEST besagt, dass der Nutzen der Anforderung klar erkennbar sein sollte (valuable)
- d) Das "T" in INVEST steht für timeboxed

3) Worauf sollte man achten, wenn man Scrum in einem Unternehmen einführen will?

- a) Das Management kann im ersten Sprint die Arbeiten des Entwicklungsteams beeinflussen
- b) Das Management sollte einen Wandel in der Unternehmensstruktur vertreten
- c) Der Scrum Master stellt das Team zusammen und hat Personalverantwortung für die Teammitglieder
- d) Der Product Owner übernimmt im ersten Projekt auch die Rolle des Scrum Masters

4) Was gehört zu den Aufgaben des Product Owners?

- a) Er schützt das Team und unterstützt es, zusammenzuwachsen
- b) Übernahme von Entwicklungstätigkeiten als normales Teammitglied
- c) Erstellung eines Releaseplans
- d) Entscheidung über die Team-Zusammensetzung

5) Welches der folgenden ist ein Scrum Artefakt?

- a) Sprint Ziel
- b) Produktkonzept
- c) Refinement Backlog
- d) Product Backlog

6) Welche der folgenden Aussagen bezüglich der Sprint Review ist falsch?

- a) Der Scrum Master lädt zur Sprint Review ein
- b) Stakeholder können ebenfalls eingeladen werden, um ungefiltert Feedback an das Team zu geben
- c) Sie findet am Ende eines Sprints statt und dauert maximal 2 Stunden
- d) Das Ziel der Sprint Review ist die Begutachtung und Abnahme der Arbeitsergebnisse eines Sprints

7) Um den Projektfortschritt besser verfolgen zu können ist es sinnvoll, weitere Metriken zu definieren. Welche der folgenden Metriken können sinnvoll sein?

- 1. Anzahl der Hindernisse**
- 2. Geplanter vs. erreichter Projektstatus**
- 3. Ergebnisse eines einzelnen Teams**
- 4. Velocity (Entwicklungsgeschwindigkeit)**

- a) Nur 2 und 4
- b) alle genannten
- c) Nur 1 und 2
- d) alle außer 1

8) Welche Eigenschaften hat nach Scrum das Product Owner Team in großen Projekten?

- 1. Es besteht mindestens aus den Product Ownern aller Teams**
- 2. Es besitzt einen Chief Product Owner, der zugleich verantwortlich für das gesamte Produkt ist**
- 3. Es kann sich selbst nach Scrum-Regeln organisieren und einen Scrum Master für das Product Owner Team benennen**
- 4. Es kann sinnvoll sein, zusätzliche Experten (z.B. einen Mitarbeiter aus dem Vertrieb) in das Product Owner Team aufzunehmen**

- a) 1, 2 und 3
- b) alle genannten
- c) Nur 1 und 2
- d) Nur 2 und 3

9) Welche der folgenden Fragen werden im Daily Scrum nicht beantwortet?

- a) Welche Aktivitäten habe ich seit dem letzten Mal abgeschlossen?
- b) Werde ich an der Ausführung meiner Aktivitäten behindert?
- c) Woran plane ich bis zum nächsten Mal zu arbeiten?
- d) Welche Anforderungen aus dem Product Backlog sind falsch abgeschätzt?

10) Welche der folgenden Aussagen ist korrekt?

- a) Der Scrum Master hat die Entscheidungsgewalt über die Anforderungen an das Produkt
- b) Das Product Backlog enthält ausschließlich die Anforderungen an einen Sprint
- c) Der Product Owner überwacht die Einhaltung der Scrum Regeln
- d) Das Sprint Backlog beschreibt eine Teilmenge der Anforderungen an das Produkt

11) Oft ist es sinnvoll, umfangreiche User Stories zu unterteilen (Splitting stories). Hierbei gilt:

- a) Splitting Stories kann auch verwendet werden, wenn das Product Backlog initial vom Scrum Master befüllt wird
- b) Durch die Unterteilungen und die damit verbundene Detailtiefe entstehen präzisere Schätzungen
- c) Splitting Stories gehört nicht zum Refinement und darf daher keine Kapazität des Entwicklungsteams beanspruchen
- d) Das Splitting Stories gehört zu den fünf Scrum Ereignissen

12) Welche der folgenden Aussagen beschreibt ein Velocity Chart?

- a) Mit dem Velocity Chart wird die Entwicklungsgeschwindigkeit des nächsten Sprints dargestellt
- b) mit Hilfe des Entwicklungsgeschwindigkeitsberichts kann direkt auf die Story Points des nächsten Sprints geschlossen werden
- c) Die ideale Kurve im Entwicklungsgeschwindigkeitsbericht ist die charakteristische Sägezahnkurve
- d) Auf der X-Achse werden die einzelnen Sprints angetragen

13) Wenn ein Projekt "Scrum of Scrums" verwendet, dann hat es folgende Eigenschaft:

- a) Scrum of Scrums findet erst ab einer Größe von 5 Scrum Teams statt
- b) Nur die Scrum Master nehmen am Scrum of Scrums teil
- c) Es findet zusätzlich zu den Daily Scrums ein weiteres Meeting von der Art eines Daily Scrums statt - das sogenannte Scrum of Scrums
- d) Das Scrum of Scrums wird in der Regel durch den Chief Product Owner moderiert

14) Das Sprint Backlog kann als Zeitmanagement-System aufgebaut sein, um den Sprint Fortschritt zu verfolgen. Welche der folgenden Eigenschaften hat das Sprint Backlog nicht?

- a) Es beinhaltet die Anforderungen und Tasks, zu denen sich das Team verpflichtet hat
- b) Es sollte mindestens einmal täglich aktualisiert werden
- c) Es ist das Ergebnis des Sprint Plannings
- d) Es enthält die vom Product Owner abgeschätzten Aufwände

15) Welche der folgenden Aussagen bezüglich Sprints ist korrekt?

- a) Im Daily Scrum werden Verbesserungsmaßnahmen umgesetzt
- b) Ein Sprint dauert maximal 30 Tage
- c) Jeder Sprint erzeugt ein eigenes, unabhängiges Produktinkrement
- d) Ziel des Sprint Plannings ist die Identifizierung von Verbesserungsmaßnahmen

16) Im Sprint Planning ist eine genaue Abschätzung der Aufwände wichtig. Welche der folgenden Aussagen ist falsch?

- a) Planning Poker ist eine Methode, um Teamentscheidungen herbeizuführen
- b) Die Fibonacci-Reihe ist eine mögliche Punktereihe für Planning Poker
- c) Der Aufwand für eine Anforderung misst sich in Scrum im Vergleich zu einer Referenzanforderung
- d) In Scrum dürfen ausschließlich Story Points (Punkte) für die Aufwände vergeben werden

17) Welche der folgenden Aussagen bezüglich des Scrum Masters ist korrekt?

- a) Übertragen auf traditionelle Projektmanagement-Methoden nimmt der Scrum Master die Rolle des Projektleiters ein
- b) Für ein effizient arbeitendes Team sollte der Scrum Master Autorität bezüglich der Arbeitsorganisation im Team haben
- c) Um die Rolle des Scrum Masters zu stärken, sollte dieser Personalverantwortung für die Teammitglieder haben
- d) Der Scrum Master trägt Verantwortung für den Scrum-Prozess und dessen korrekte Implementierung

18) Wer ist kein Mitglied des Scrum Teams?

- a) Scrum Master
- b) Stakeholder
- c) Product Owner
- d) Entwicklungsteam

19) Im Release Management sind die Zusammenhänge der verschiedenen Planungsebenen wichtig. Welche der folgenden Aussagen ist falsch?

- a) Bei Projekten, die weniger als 6 Sprints dauern, benötigt man generell keine Release Planung
- b) Ein Release besteht aus mindestens einem Sprint
- c) Ein Task muss in einem Sprint erledigt werden können
- d) Ein Task ist einer Story zugeordnet

20) Die Priorisierung des Product Backlogs durch den Product Owner ist ein wichtiges Mittel, um das Projekt zu steuern. Bei der Priorisierung sollte beachtet werden

- a) dass Anforderungen mit einem hohen Risiko und hohen Nutzen möglichst erst am Ende umgesetzt werden sollten
- b) dass die wichtigsten Anforderungen nicht sofort umgesetzt werden
- c) dass Anforderungen mit einem hohen Risiko und hohen Nutzen möglichst früh umgesetzt werden sollten
- d) dass Anforderungen mit einem hohen Risiko und geringen Nutzen als erstes umgesetzt werden sollten

21) Welche der folgenden Aussagen trifft auf Scrum zu?

- a) Aus dem Product Backlog werden Anforderungen in das Sprint Backlog übernommen
- b) Der Product Owner überwacht die Daily Scrums
- c) Bei Unstimmigkeit wird das Scrum Mediation Team eingeschaltet
- d) Ein Sprint dauert in der Regel 2-4 Tage

22) Bei komplexen Systemen kann es sinnvoll sein, ein Produktkonzept zu erstellen. Welche der folgenden Aussagen treffen zu?

- a) Aus der Produktidee wird zunächst ein Produktkonzept formuliert, bevor das Product Backlog gefüllt wird
- b) Das Produktkonzept beschreibt ausschließlich den größten Mehrwert des Produkts
- c) Ein Produktkonzept wird nur in großen und verteilten Projekten genutzt
- d) Nachdem das Product Backlog initial befüllt wurde, sollte man die zentralen Aspekte und Eigenschaften in einem Konzept zusammenfassen

23) Wer erstellt das Produktinkrement?

- a) Der Product Owner
- b) Das Scrum Team
- c) Der Scrum Master
- d) Das Entwicklungsteam

24) Was wird in Scrum nicht vom Entwicklungsteam entschieden?

- a) Welche Anforderungen es für die Umsetzung in einem Sprint plant
- b) Wo und zu welcher Zeit das Daily Scrum stattfinden soll
- c) Welche Arbeitsergebnisse im Sprint Review abgenommen werden
- d) Welche Regeln für alle Teammitglieder gelten

25) Welcher der folgenden Aspekte ist ein zentraler Wert des agilen Manifests?

- a) Effizienter Einsatz von Projektmanagement-Tools
- b) Striktes Einhalten von Projektplänen
- c) Der Mensch steht im Mittelpunkt der Softwareentwicklung
- d) Vertragsverhandlungen müssen zu detaillierten Verträgen führen

26) Es gibt verschiedene Varianten, die Velocity (Entwicklungsgeschwindigkeit) zu ermitteln. Welche der folgenden Aussagen ist korrekt?

- a) Die Entwicklungsgeschwindigkeit sollte von Sprint zu Sprint um mindestens 20% ansteigen
- b) Die Entwicklungsgeschwindigkeit ist unabhängig von der Größe des Entwicklungsteams
- c) Die Entwicklungsgeschwindigkeit wird in der Sprint Retrospektive für den folgenden Sprint festgelegt
- d) Wenn vor dem ersten Sprint bereits ein Releaseplan existieren muss, kann die Entwicklungsgeschwindigkeit mit Hilfe des Teams abgeschätzt werden

27) Welche der folgenden Aussagen bezüglich eines Entwicklungsteams in Scrum ist korrekt?

- a) Ein Entwicklungsteam sollte mehr als 20 Mitglieder haben
- b) Detaillierte Fragen zu Anforderungen werden in Scrum vom Entwicklungsteam direkt mit Kunden ohne Einbindung des Product Owners geklärt
- c) Effizienz spielt im Entwicklungsteam kaum eine Rolle, da nur Effektivität bewertet wird
- d) Ein Entwicklungsteam sollte so autonom arbeiten können, dass das Sprint-Ziel ohne externe Abhängigkeiten erreicht werden kann

28) Welche der folgenden Eigenschaften gilt nicht für ein Produktinkrement in Scrum?

- a) Es ist lauffähig
- b) Es muss lauffähig, aber nicht zwangsläufig bereits dokumentiert sein
- c) Es ist getestet
- d) Es ist dokumentiert

29) Bis ein Scrum Team optimal und effizient arbeitet, durchläuft es einen längeren Teamprozess. Dabei gilt:

- a) Nach Tuckman folgt auf die Phase des "Forming" unmittelbar die Phase des "Performing"
- b) Die Phase des Zusammenfindens wird nach Tuckman als "Finding" bezeichnet
- c) Nachdem sich das Team auf gemeinsame Arbeitsweisen und Normen geeinigt hat, gelangt es in die Auseinandersetzungsphase
- d) Die Auseinandersetzungsphase wird nach Tuckman auch als "Storming" bezeichnet

30) Welche der folgenden ist ein Wert in Scrum (Scrum Values)?

- a) Flexibility
- b) Empiricism
- c) Agility
- d) Courage

Punkte:	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Punkte:	a	b	c	d
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

Punkte:	a	b	c	d
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Antwortbogen

Datum:

Ort:

VA-Nr.:

Teilnehmerdaten (bitte in Blockschrift ausfüllen)

ID:

P									
---	--	--	--	--	--	--	--	--	--

Anrede: ☐ Frau ☐ Herr

Name: _____

Vorname: _____

Geburtsdatum: _____

Geburtsort: _____

Unterschrift: _____

Ergebnis:

_____ von 30 maximal erreichbaren Punkten

entspricht _____ %

☐ Prüfung bestanden

☐ Prüfung nicht bestanden

Prüfungsbeauftragter



Bestehensgrenzen

Maximal erreichbare Punkte 30
 Bestanden ab 66 %

min. 20 Punkte

Punkte	Prozent	Punkte	Prozent	Punkte	Prozent
30	100 %	min.--> 20	66 %	10	33 %
29	96 %	19	63 %	9	30 %
28	93 %	18	60 %	8	26 %
27	90 %	17	56 %	7	23 %
26	86 %	16	53 %	6	20 %
25	83 %	15	50 %	5	16 %
24	80 %	14	46 %	4	13 %
23	76 %	13	43 %	3	10 %
22	73 %	12	40 %	2	6 %
21	70 %	11	36 %	1	3 %
min.--> 20	66 %	10	33 %		



	a	b	c	d
1			☒	
2			☒	
3		☒		
4			☒	
5				☒
6			☒	
7		☒		
8		☒		
9				☒
10				☒

	a	b	c	d
11		☒		
12				☒
13			☒	
14				☒
15		☒		
16				☒
17				☒
18		☒		
19	☒			
20			☒	

	a	b	c	d
21	☒			
22	☒			
23				☒
24			☒	
25			☒	
26				☒
27				☒
28		☒		
29				☒
30				☒

Lösungsbogen

Datum:

Ort:

VA-Nr.:

