



Go Scrumming

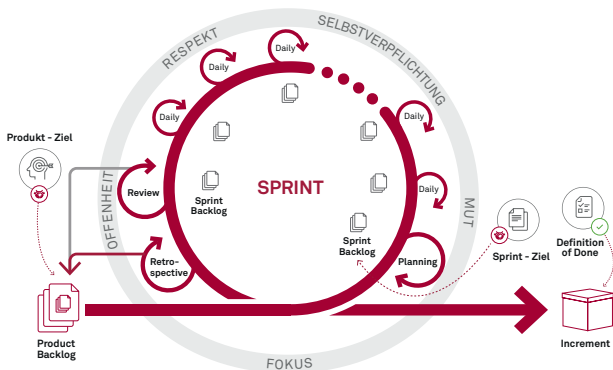
Scrum Tools für die Hosentasche

value – inspired by people

•msg

Go Scrumming

Scrum Tools für die Hosentasche



Inhalt

6	Herzlich willkommen beim „Scrummen“
9	Scrum auf dem Bierdeckel
10	Die Definition
11	Und was heißt das?
13	Bitte?!?
16	Das Agile Manifest
17	Die 3 Säulen von Scrum
21	Die Scrum Werte
24	Selbstorganisation
25	Vertrauen, Courage und Transparenz
27	Scrum Verantwortlichkeiten
29	Entwickler
30	Product Owner (PO)
32	Die Ereignisse in Scrum
33	Der „Sprint“
34	Sprint Planning
36	Sprint Planning – Teil 1
37	Sprint Planning – Teil 2
38	Sprint Planning – Teil 3

39	Sprint-Ziel
41	Daily Scrum
43	Sprint Review
45	Sprint Retrospektive
48	Refinement
51	Refinement Poker
53	Release-Planung
56	Impediment Backlog
57	Die „Definition of Done“ (DoD) – „Fertig!“
59	Die „Definition of Ready“ (DoR) – „Fertig!“ des POs
62	Product Backlog
64	Increments
65	Scrum Good Practices
66	Sprint Burndown Chart
69	Kleines Scrum Vokabular
70	Scrum Timeboxen
71	Software Tools
72	Glossar
77	Der Autor

Herzlich willkommen beim „Scrummen“

Du hältst ein Nachschlagewerk zu Scrum in der Hand, in welchem die Grundlagen zu Scrum dargestellt und erläutert werden.

Den ein oder anderen Tipp, sowie Informationen darüber, was hinter Scrum und dem agilen Vorgehen steckt, findest du ebenso in dieser kleinen Broschüre.

Dieses „**Hosentaschen**“-Nachschlagewerk ist für alle an Scrum Interessierten gedacht, soll Einblicke in das Scrum Framework geben, dessen Praktiken vermitteln und Dir beim täglichen „**Scrummen**“ dienen.

Voraussetzung sollte sein, dass Dir die Scrum Grundlagen, eventuell aus einem Scrum Kurs, schon bekannt sind, damit Dir dieses Buch auch wirklich ein Hilfsmittel ist, und Dir die Erklärungen helfen können.

Im Glossar am Ende des Buches werden Kürzel erläutert, und in den Fußnoten werden zusätzliche Erläuterungen gegeben.

„Noch ein Buch über Scrum“ fragst du dich? Ja, denn kein anderes Vorgehensmodell der agilen Welt bietet einem so viel Handlungsspielraum, bei so viel Agilität und so wenig „Vorgaben“.

Es basiert auf dem offiziellen Scrum Guide¹ und der jahrelangen Erfahrung, welche ich mit Scrum und der Einführung von Scrum in den unterschiedlichsten Unternehmen und Bereichen gemacht habe.

Dabei werde ich natürlich keine „Scrum Bibel“ für dich bereithalten, denn das widerspräche dem Grundgedanken und Wesen von Scrum als einem empirischen Vorgehen, welches sich stetig weiterentwickelt.

Ich werde in diesem Buch weitestgehend davon Abstand nehmen, dass Scrum ein Prozess² ist. Scrum ist ein Rahmenwerk, innerhalb dessen man sich frei bewegen kann, was die Sache an sich sehr schwer macht. Scrum Master sind dafür da, die Einhaltung dieser Rahmenbedingungen sicherzustellen, sodass sich diese „Freiheit“ wirklich entfalten kann, und zum Erfolg führt.

In diesem Buch werden Good Practices zu Scrum erklärt und erläutert, welche jedoch nicht zu Scrum gehören, bzw. nicht von Scrum definiert sind, sondern sich über die Jahre hinweg als sehr hilfreich im Umgang mit Scrum erwiesen haben und Scrum dadurch erfolgreicher machen.

1 Scrum Guide: <http://www.scrumguides.org/index.html>

2 Ein Prozess, klassischer deterministischer Definition, ist eine Abfolge von Zustandsänderungen, bei dem jeder Zustand kausal von anderen, vorherigen abhängig ist und von diesen bestimmt ist. Scrum ist in seiner Definition eben kein solcher „Prozess“, sondern gibt vielmehr Ereignisse/Best Practices vor, welche dem speziellen Zweck dienen, Projekte effektiver und effizienter zum Erfolg zu führen.

Scrum auf dem Bierdeckel

Wenn dich jemand fragt: „Was ist Scrum?“, dann nimm einfach einen Stift und einen Notizzettel oder Bierdeckel, und male folgendes Bild, mit den Worten: „Das ist Scrum!“:



Die Definition

„Scrum ist ein leichtgewichtiges Rahmenwerk, welches Menschen, Teams und Organisationen hilft, Wert durch adaptive Lösungen für komplexe Probleme zu generieren.“

Diese Definition stammt aus der offiziellen deutschen Übersetzung des Scrum Guides, welcher als Regelwerk für Scrum verstanden wird. Wichtig hierbei ist, dass es sich bei Scrum um ein **Framework** handelt und **nicht** um eine formale **Prozessbeschreibung**.

Die Menschen, das Individuum als solches und Ganzes, ihre Interaktionen und ihre Kreativität sowie die bestmögliche Zusammenarbeit im Team steht im Mittelpunkt, nicht die Einhaltung von Prozessen.



3 Scrum Guide: <https://scrumguides.org/>

Und was heißt das?

Nun, ganz einfach: Scrum gibt ein paar wenige Regeln und Artefakte vor, an die man sich halten muss und steckt dadurch einen Rahmen ab, innerhalb dessen man sich frei bewegen kann.

Es wird also nicht vorgegeben, welche Dokumente in welcher Reihenfolge auszufüllen, oder welche Templates zu benutzen sind, damit ein Projekt oder Produkt erfolgreich wird. Weit gefehlt. Sollten dies Anforderungen innerhalb einer Organisation sein, so hat diese entsprechend - unter Einhaltung des Rahmenwerkes - diese selbst auszuarbeiten und bereitzustellen.

Aus der Definition von Scrum kannst du entnehmen, dass Scrum den Anspruch hat **„Produkte mit höchstmöglichem Wert auszuliefern“**. Um diesen höchstmöglichen Geschäftswert für ein Produkt zu erreichen, bedient sich Scrum der **„empirischen Prozesssteuerung“**.

Der Kern der empirischen Prozesssteuerung ist die Tatsache, dass Wissen aus Erfahrung gewonnen wird, sowie die kontinuierliche Inspektion und Adaption, die in Scrum zum Tragen kommt und gelebt wird. Die Voraussetzungen hierfür sind Transparenz, welche durch verschiedene Artefakte und Meetings in Scrum gefordert und gefördert wird, sowie Vertrauen und Mut (Courage⁴).

Denn nur durch Vertrauen und Courage kann Transparenz erzeugt werden.⁵

Das menschliche Potenzial jedes Einzelnen wird mit Selbstorganisation, unbürokratischem Vorgehen, Disziplin und Wertschätzung am intensivsten genutzt. Transparenz und Flexibilität zeichnen das Vorgehen aus.

4 Auch in der deutschen Literatur über Scrum wird sehr häufig das Wort und der Begriff „Courage“ verwendet. Auch wenn dies in direkter Übersetzung „Mut“ bedeutet, so hat sich aus der Historie von Scrum und dessen Literatur heraus das Wort „Courage“ etabliert, weswegen in diesem Buch sowohl „Mut“ als auch „Courage“ in gleicher Bedeutung Platz finden.

5 Mehr Informationen zu „**Transparenz, Vertrauen und Courage**“ auf Seite 21

Bitte?!?

Scrum ist, ganz kurz und einfach gesagt, ein einfaches Rahmenwerk (Framework), welches nur sehr wenige Regeln und Artefakte vorsieht.

Innerhalb dieses Rahmenwerks, in das die Entwicklungsteams selbstbestimmt selbsterprobte Praktiken einbetten, in das die Scrum Teams können diese wiederum kreativ & produktiv arbeiten.

Scrum sieht einen evolutionären, empirischen Ansatz vor, innerhalb dessen iterativ vorgegangen wird.

„Empirisch“ – In Scrum wird nicht nach theoretischen Ansätzen geplant, sondern es werden die Erfahrungswerte der Personen herangezogen, welche auch die entsprechenden Aufgaben umsetzen.⁶

„Iterativ“ – Die Termine bzw. Zeitfenster (Timeboxen), also die Sprints, in welchen gearbeitet wird, sind fest, nicht verschiebbar und wiederholen sich auf Basis fester Zeitzyklen iterativ. Am Ende eines Sprints liegt den Kunden und dem Product

Owner ein verwendbares und wertvolles Increment vor, welches diesen zur Inspektion und Adaption des Inkrements dient (Inspect & Adapt).

Die Termine der Sprints sind nicht verschiebbar, sondern fix. Die Menge an Anforderungen, welche pro Sprint abgearbeitet werden können, ist jedoch flexibel.

„Evolutionär“ – Dadurch, dass in Scrum in kurzen Zeitabschnitten gearbeitet wird, wird vor jedem neuen Sprint auch neu geplant. Durch das Gelernte und durch das erweiterte Wissen der vorherigen Sprints fließen diese Erfahrungswerte in die Planung der kommenden Sprints mit ein. Dies hat zur Folge, dass sich sowohl Prioritäten, als auch das Pensum, was ein Team in einem Sprint abarbeiten kann, ändern können. Dies schafft ein Höchstmaß an Flexibilität.

Da Sprints immer fix in ihrer Länge sind, entwickelt das Scrum Team über die Zeit hinweg zunehmend Erfahrung darin, wie viele

Arbeitseinheiten/Anforderungen in einem Sprint erledigt werden können. Damit wird die Planung in einem Team mit zunehmender Anzahl der Sprints immer verlässlicher und treffsicherer.

Hinzu kommt, dass das Scrum Team aus dem bereits Geleisteten (Erfolge und Misserfolge) lernt, und sich somit verbessert und schneller wird.

Evolutionär heißt auch, dass sich die Teams selbst organisieren. Sich selbst organisierende Teams entwickeln die höchste Produktivität, in der das Ganze mehr ergibt, als die Summe aller Teile. Scrum stellt Ideen, Ansätze, Prozesse und Methoden zur Verfügung, um Selbstorganisation zu fordern und zu fördern.

7 PSI: „Potentially Shippable (Product) Increment ist das Produkt, welches am Ende eines Sprints, einer Iteration vorliegt, und potenziell dem Kunden ausgeliefert werden kann.

„Das Manifest für agile Softwareentwicklung“⁸

Individuen & Interaktionen	sind wichtiger als	Prozesse & Tools
Funktionierende Software	ist wichtiger als	Umfangreiche Dokumentation
Kooperation mit Projektbetroffenen	ist wichtiger als	Vertragsverhandlungen
Reaktion auf Änderungen	ist wichtiger als	Verfolgung eines festgelegten Planes

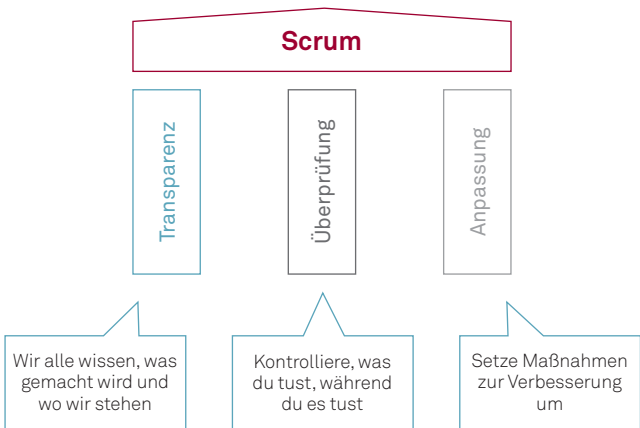
Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, messen wir den Dingen auf der linken Seite größeren Wert bei.

Scrum baut auf diesen Werten auf, ohne welche Scrum in seiner Form und seinem Vorgehen nicht funktionieren würde.

⁸ <http://www.agilemanifesto.org/>

Die 3 Säulen von Scrum

Scrum ist ein inkrementeller, iterativer Ansatz, der auf drei Säulen ruht:⁹



⁹ scrum.org, Scrum Guide, 1991 - 2021 Ken Schwaber and Dr. Jeff Sutherland

Die erste Säule, ist die Transparenz (Transparency):

Alle Aspekte des Vorgehens, die das Ergebnis beeinflussen, sind sichtbar. Ebenso muss das, was gezeigt und gesehen wird, realistisch sein. Dies betrifft nicht nur das Increment (siehe Seite 64) an sich, sondern auch Artefakte, wie z.B. Aussagen über Planungen, das Sprint Burndown Chart sowie Aussagen über PBIs (siehe Seite 56), usw.

Die zweite Säule ist die Überprüfung (Inspection):

Regelmäßige Überprüfungen ermöglichen das frühzeitige Erkennen von eventuellen (inakzeptablen) Abweichungen. Die Häufigkeit der Kontrollen sollte die Toleranz der Überprüfungen nicht überschreiten (so wenig wie möglich, so viel wie nötig).

In Scrum überprüfen wir uns stetig, spätestens jedoch im Daily Scrum, bei welchem wir unsere Arbeit der vergangenen 24 Stunden überprüfen, sowie im Sprint Review, innerhalb dessen wir das Increment und die Arbeit des vergangenen Sprints überprüfen.

In der Sprint Retrospektive werden das Vorgehen, die Tools, eventuelle übergeordnete Prozesse und unser Verhalten überprüft.

Die dritte und letzte Säule ist die Anpassung (Adaptation):

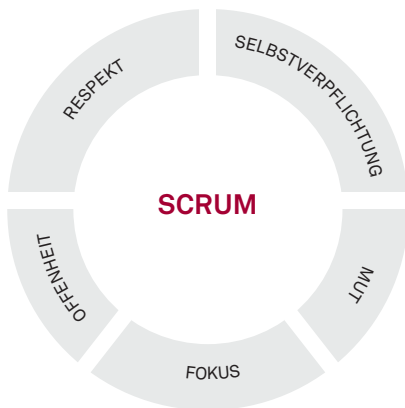
Erkennt die Inspektion beim Vorgehen, dem Prozess, dem Verhalten, den verwendeten Materialien oder den gelieferten Ergebnissen Abweichungen außerhalb der zulässigen Grenzen, so müssen diese so schnell wie möglich behoben werden, um weitere Abweichungen zu verhindern.

Dies betrifft nicht nur die Kernelemente einer Gesamtentwicklung, oder der aktuellen Entwicklungsphase (Sprint), sondern das gesamte Vorgehen an sich. Werden z.B. Defects während der laufenden Entwicklung entdeckt und betreffen diese ein in Arbeit befindliches Feature, so müssen diese umgehend behoben werden. Adressiert ein gefundener Defect eine andere Funktion, so wird dieser Defect in den Backlog aufgenommen, durch das Team und den PO im Product Backlog einsortiert, bzw. priorisiert, und durch die kurzen Zyklen des Vorgehens schnellstmöglich abgearbeitet.

Wird z.B. während der Entwicklung oder der laufenden Analyse von Anforderungen (PBIs) identifiziert, dass die aktuelle Softwarearchitektur nicht mehr oder teilweise nicht mehr zukunftssicher ist oder neue Features mit dieser nicht mehr realisiert werden können, so kann kurzfristig, frühestens jedoch im kommenden Sprint auf diese Änderung reagiert werden.

„**Inspect & Adapt**“ betrifft in Scrum den LifeCycle aller Dinge, welche passieren: Entwicklung, Prozesse, Vorgehen, Verhalten, Aussagen,

Die Scrum Werte



„Commitment“ (Selbstverpflichtung / Engagement / Einsatz):

... zum Team ... zur Qualität ... zur Zusammenarbeit ... stetig zu lernen ... jeden Tag immer wieder das Beste zu geben, was möglich ist ... zum Sprintziel ... zur Professionalität ... zur Selbstorganisation ... zur Leistung ... zu den agilen Grundprinzipien ... auslieferbare und funktionierende Software auszuliefern ... stetig nach Verbesserungen zu suchen ... zur Definition of Done (DoD) ... zum Scrum Framework ... Werte auszuliefern ... Arbeit fertig zu stellen (bevor neue Arbeit begonnen wird) ... zur kontinuierlichen Inspektion und Adaption (Inspect & Adapt) ... zur Transparenz!

Fokus

... auf das, was du weißt (vermeide YAGNI – „You Ain't Gonna Need It“) ... auf das, was am Wichtigsten ist ... auf das zeitlich Naheliegendste ... auf deine Arbeit ... auf das Einfachste was funktionieren wird auf das Produktziel

Offenheit

Sei transparent. Sei offen gegenüber dem, was du tust und deiner Arbeit. Sei offen den Menschen/Kollegen gegenüber. Sei offen und

bereit zur übergreifenden Zusammenarbeit bzgl. der Fähig-, Fertigkeiten und Disziplinen. Sei offen zur Zusammenarbeit mit Stakeholdern und dem weiteren Umfeld. Sie offen gegenüber Änderungen.

Respekt

Respektiere die Erfahrung der Menschen, ihr persönliches Umfeld und ihren Hintergrund. Respektiere die Mannigfaltigkeit und Vielfältigkeit (es macht dich erfolgreich und besonders). Respektiere andere Meinungen (du kannst nur daraus lernen).

Mut

Zeige Mut, indem du nicht Dinge machst und baust, welche keiner will und welche nicht benötigt werden, indem du hinterfragst und offen Anforderungen in Frage stellst. Zeige Mut, Software, die nicht fertig ist, nicht auszuliefern. Zeige Mut, indem du alle Informationen preis gibst und zur Verfügung stellst (Transparenz). Zeige Mut, indem du zugibst, nicht perfekt zu sein (keiner ist es!). Zeige den Mut, den Weg und die Richtung zu wechseln. Zeige deinen Mut, indem du ehrlich bist.

Selbstorganisation

Im Konzept von Scrum steht die Selbstorganisation des Scrum Teams im Fokus und im Mittelpunkt. Dies bedeutet, dass das Scrum Team bei allen Planungssitzungen anwesend sein und im Mittelpunkt der Planungen und Aufwandsschätzungen steht.

Die Schätzungen des Umfangs von Aufgaben (meist in Angaben von relativen empirischen Zahlen) werden stets gemeinsam im Team durchgeführt. Dazu wird in Scrum häufig Planning Poker¹⁰ verwendet. Da Scrum nur einen Rahmen aufzeigt, verbessert das Scrum Team das Vorgehen und seine Prozesse¹¹ im Detail ständig, um die Scrum Team Produktivität zu steigern (siehe auch „Sprint Retrospektive auf Seite 44“).

Die Aufgabe des Scrum Masters ist hier die des Moderators. Er hat sicher zu stellen, dass die beiden anderen Ergebnisverantwortlichkeiten (Entwickler & Product Owner) die erforderlichen Maßnahmen treffen, diese und ihre Aufgaben effektiv und effizient durchzuführen.

10 Planning Poker: Definiert und erfunden von James Grenning 2002, angepasst und als eingetragenes Markenzeichen registriert durch Mike Cohn.

11 Als Prozess ist hier nicht Scrum gemeint, sondern die eigenen Prozesse des Teams z.B. um sich selbst zu organisieren oder seine Produktivität zu verbessern, etc.

Vertrauen, Mut und Transparenz

In Verbindung mit den agilen Werten und Prinzipien sowie im Zusammenhang mit Scrum wird sehr oft „**Vertrauen**“, „**Mut**“ und „**Transparenz**“ erwähnt und dass es ohne Vertrauen und Mut keine Transparenz gibt.

Warum dies so ist, lässt sich sehr schön anhand der folgenden Grafik erläutern:



Um ein Ziel erfolgreich realisieren zu können, brauche ich häufige „**Inspektion**“ & „**Adaption**“.

Diese erreiche ich aber nur durch vorbehaltlose „**Transparenz**“.
Transparenz hingegen erreiche ich nur durch „**Mut**“ und „**Vertrauen**“.

Vertrauen, dass jemand für den Mut etwas zu sagen, zu zeigen, zu machen oder auch nicht zu machen, nicht „bestraft“ wird. Nur dadurch wird vorbehaltlose Transparenz erzeugt.

Ergebnisverantwortlichkeiten in Scrum

Innerhalb des Scrum Frameworks gibt es genau drei Ergebnisverantwortlichkeiten:¹²



- „Product Owner“



- „Scrum Master“



- „Entwickler“

Alle drei zusammen ergeben das „Scrum Team“. Es gehören **nur diese** drei Ergebnisverantwortlichkeiten zu einem Scrum Team, nicht mehr und nicht weniger. Im Folgenden werden diese Rollen kurz erklärt.

12 Weitere Rollen, wie z.B. die eines Projektmanagers, Projektleiters oder Key-Product Owners sind in Scrum nicht vorgesehen, können jedoch unter speziellen Umständen, und mit speziellen Linien- & Organisationsverantwortlichkeiten je nach Projekt und Unternehmen nötig sein.



Scrum Master (SM)

Der Scrum Master wird im Allgemeinen Sprachgebrauch, auch wenn Scrum selbst kein Prozess ist, als „Prozess Owner“ für Scrum verstanden, der dafür sorgen muss, dass das Scrum **Vorgehen** von allen richtig verstanden, korrekt angewandt und eingehalten wird.

Er muss sicherstellen, dass das Team die Werte, Praktiken und Regeln von Scrum einhält. Zudem hilft er bei der Einführung von Scrum. Er zeigt dem Team durch Coaching und Mentoring, wie es bessere Ergebnisse erreichen kann. Desweiteren beseitigt er Hindernisse für das Team (Impediments), die das Entwicklungsteam am idealen Vorankommen hindern.

Der Scrum Master ist ein „**Servant Leader**“, eine „**dienende Führungskraft**“ und ist eine **Management Rolle**.

Er dient dem Product Owner, dem Entwicklungsteam als auch der Organisation.

Den **Product Owner** unterstützt er z.B. darin, neue Techniken zur effektiven Verwaltung des Product Backlogs, zu etablieren.

Die gesamte **Organisation**, aber auch Kunden und einzelne Projekte, unterstützt er dadurch, dass er sie bei der Umsetzung von Scrum führt, coacht und somit den Nutzen von Scrum maximiert. Dies geschieht unter anderem durch die kontinuierliche Inspektion und Adaption aller Abläufe, welche dadurch stetig verbessert werden und optimal ablaufen. Weiter ist er dafür zuständig, den agilen Gedanken und Scrum weiter in das Unternehmen hineinzutragen, um somit das Unternehmen, die Organisation auf einem Weg in ein agiles Unternehmen zu führen (Nur ein Beispiel und eine Aussicht auf eine unternehmensweite Ausweitung von Scrum).

Die Entwickler unterstützt er unter anderem dabei, neue Techniken und Praktiken zu etablieren, um ihre eigenen Prozesse zu verbessern, Hindernisse (Impediments), welche das Team selbst nicht beseitigen bzw. lösen kann, zu lösen, und durch gezielte und erfahrene Praktiken effektive und Erfolg bringende Retrospektiven zu moderieren.

Der Scrum Master darf niemals zugleich Product Owner sein, da dies einen Interessenkonflikt darstellen würde.



Entwickler

Die Entwickler bestehen aus 6 ± 3 Mitarbeitern, welche gemeinsam dazu in der Lage sind, die Einträge aus dem Produkt Backlog in den Sprint Backlog zu überführen und dieses selbstorganisiert in ein auslieferbares Produktinkrement (**PSI**) umzusetzen.

Das Team ist interdisziplinär besetzt und muss in der Lage sein alle notwendigen Arbeiten, die zur Umsetzung der Product Backlog Items notwendig sind, durchzuführen. Daher bedarf es keiner Subteams wie „Tester“, „Analysten“ oder „Architekten“, welche für Teile der Arbeit zuständig sind. Die Entwickler sind stets als Ganzes für das Ergebnis verantwortlich.



Product Owner (PO)

Er muss den Wert bzw. den Nutzen der Teamarbeit maximieren (ROI¹³) und die Kosten der Entwicklung unter Betrachtung aller Aspekte optimieren (TCO¹⁴).

Der Product Owner ist für das Management des Product Backlogs sowie für das gesamte Requirements- Management/Engineering verantwortlich und hat das letzte Wort über die Ordnung der umzusetzenden Anforderungen (Product Backlog Items, PBIs).

- 13 Bei einer Berechnung des ROI (Return on Invest) bezogen auf eine Einzelinvestition wird grundsätzlich davon ausgegangen, dass die Rückflüsse der Investition aus einer vorangegangenen Analyse bereits bekannt sind.

$$\text{ROI} = \frac{\text{Gewinnanteil}}{\text{Kapitaleinsatz}}$$

Grundsätzlich ist eine Berechnung des ROI immer nur dann interessant, wenn die Investition auch zum Unternehmenserfolg beitragen kann, d.h. dass eine Amortisation innerhalb der Nutzungsdauer erreicht wird.

- 14 Total Cost of Ownership (TCO, Gesamtbetriebskosten) ist ein Abrechnungsverfahren, das Verbrauchern und Unternehmen helfen soll, alle anfallenden Kosten von Investitionsgütern (wie beispielsweise Software und Hardware in der IT) abzuschätzen. Die Idee dabei ist, eine Abrechnung zu erhalten, die nicht nur die Anschaffungskosten enthält, sondern alle Aspekte der späteren Nutzung (Energiekosten, Reparatur und Wartung) der betreffenden Komponenten. Es gibt ca. 20 verschiedene Modelle der Berechnung und Auswertung sowie über die Kennzahlen des TCO. Wichtig ist jedoch die direkte Gegenüberstellung und Auswertung direkter- und indirekter Kosten.

Er stellt sicher, dass der Product Backlog für jeden zugänglich ist. Der PO priorisiert/ordnet den Backlog und hat dafür Sorge zu tragen, dass alle PBIs so formuliert sind, dass sie von allen Beteiligten klar verstanden werden. (nicht nur die Form, z.B. als User Story ist hier wichtig, sondern auch der klar verständliche Inhalt)

Des Weiteren steht der PO dem Team jederzeit Rede und Antwort, wenn es um Klärungsbedarf bezüglich der Product Backlog Items geht.

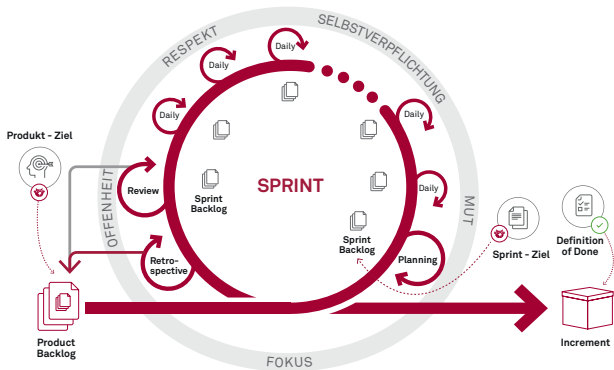
Der **Product Owner** ist eine **einzelne Person, kein Komitee und keine Abteilung**. Andere, wie z.B. Kunden, Auftraggeber, Geschäftsführung, und jedliche Stakeholder, können höchstens auf den PO Einfluss nehmen, um das Backlog indirekt zu gestalten.

Der Product Owner ist das „**single wringable neck**“, ein Ausdruck dafür, dass der **Product Owner vollverantwortlich** für den Projekterfolg ist.

Die Ereignisse in Scrum

In Scrum gibt es, wie eingangs schon erwähnt, wenige, aber dafür klare und einfache Ereignisse. Diese sind:

- Sprint (als Container)
- Sprint Planning (als Planungseinheit)
- Daily Scrum (als Planungseinheit)
- Sprint Review (Inspect & Adapt sowie als Planungseinheit)
- Sprint Retrospektive (Inspect & Adapt)





Der „Sprint“

In Scrum wird der „**Container**“, innerhalb dessen sich alle Ereignisse, alle Aktivitäten und die Arbeit abspielen, als „**Sprint**“ bezeichnet. Ist ein Sprint zu Ende, so beginnt unmittelbar danach der nächste Sprint.

Der Sprint beginnt mit dem „**Sprint Planning**“ Meeting und endet mit dem „**Sprint Review**“ und der „**Sprint Retrospektive**“ als letztes Meeting im Sprint.

Danach wird sofort von vorne begonnen. **Alle Aktivitäten** finden **innerhalb** des **Sprints** statt. Eine „Sprint Pause“ (Sprint Gap), wie fälschlicherweise von vielen angewandt um verwaltungstechnische Aufgaben und Arbeiten zu erledigen, ist nicht vorgesehen und nicht nötig, und hat mit Scrum in seiner Reinform nichts mehr zu tun.

Sprints sollten in ihrer „Länge“ 30 Kalendertage nicht überschreiten.

Mehr Informationen im Kapitel „**Scrum TimeBoxen**“.



Sprint Planning

Jeder Sprint beginnt automatisch mit dem **“Sprint Planning“** Meeting.

Der Product Owner eröffnet das Sprint Planning Meeting mit der Vorstellung eines Sprintziels, welches inhaltlich zu den vorselektierten PBIs passt, jedoch auch während des Meetings mit dem Team gemeinschaftlich definiert werden kann und sich auf Grund der vorgestellten Anforderungen oder auch eines eventuell gegebenen Releaseziels ergibt.

Der Product Owner stellt die Product Backlog Items „User Stories“ vor und bespricht diese mit den Entwicklern.

In diesem Meeting wird der kommende Sprint geplant und geschätzt. Unter Berücksichtigung eines eventuell schon vorgegebenen Sprintzieles selektiert das Team so viele Aufgaben, wie sie innerhalb eines Sprints „fertig“ (“Done” → Definition of Done) stellen können. Nur das Team definiert, wie viele Product Backlog Items es pro Sprint umsetzt.

Wichtige Voraussetzungen bzw. wichtiger Input für das Sprint Planning Meeting sind der geordnete Product Backlog, das letzte Increment, die Kapazität der Entwickler sowie die letzte Performance (Velocity) des Scrum Teams.

Im Sprint Planning werden die folgenden drei Themen behandelt:

Thema Eins: Warum ist dieser Sprint wertvoll?

Hier arbeitet das gesamte Scrum Team zusammen, um ein Sprint-Ziel zu definieren. Das Sprint-Ziel muss vor dem Ende des Sprint Planning feststehen.

Thema Zwei: Was kann in diesem Sprint abgeschlossen (Done) werden?

Hier arbeiten die Entwickler mit dem Product Owner zusammen, um den Umfang des Sprint zu konkretisieren.

Thema Drei: Wie wird die ausgewählte Arbeit erledigt?

Die Entwickler planen für jedes ausgewählte Product Backlog Item die notwendige Arbeit, um ein Increment zu erstellen, das der Definition of Done entspricht.



Spring Planning - Thema Eins: Warum ist der Sprint wertvoll?

Der Product Owner stellt vor, wie das Produkt in diesem Sprint seinen Wert und seinen Nutzen erhöhen kann.

Das gesamte Scrum Team arbeitet anschließend zusammen daran, ein Sprint-Ziel zu definieren, das deutlich macht, warum der Sprint für die Stakeholder wertvoll ist.

Das Sprint-Ziel muss vor dem Ende des Sprint Planning fertig ausgearbeitet werden.

15 Mehr Informationen zu den Sprint-Events und deren Dauer im Kapitel „**Scrum Timeboxen**“ auf Seite 65



Sprint Planning - Thema Zwei: Was kann im Sprint fertig werden?

Durch Diskussion mit dem Product Owner wählen die Entwickler Product Backlog Items aus, die in diesen Sprint aufgenommen werden sollen.

Die zu erledigenden User Stories oder Anforderungen können dabei noch verfeinert werden, was das Verständnis und die Zuversichtlichkeit erhöht.

Sind die vorgestellten User Stories noch nicht geschätzt worden, so werden diese Anforderungen hier, z.B. mit Verwendung von Planning Poker, vom Scrum Team geschätzt.

Einzuschätzen, wieviel innerhalb eines Sprints fertiggestellt werden kann, kann herausfordernd sein. Je mehr die Entwickler jedoch über ihre Geschwindigkeit in der Vergangenheit (Velocity), ihre künftige Kapazität und ihre Definition of Done wissen, desto zuversichtlicher werden sie in ihren Sprint Forecasts sein.



Sprint Planning - Thema Drei: Wie wird die Arbeit erledigt?

Für jedes ausgewählte PBI planen die Entwickler die erforderliche Arbeit, um ein Increment zu erstellen, das die Definition of Done erfüllt. Dies wird oft dadurch erledigt, dass PBIs in kleinere Arbeitseinheiten von einem Tag Dauer oder kürzer zerlegt werden. Ob und wie das gemacht wird, ist die alleinige Entscheidung der Entwickler. Niemand sonst schreibt ihnen vor, wie PBIs in wertvolle Inkremente umgewandelt werden. Die Entwickler verwenden hierzu ihre eigenen Tools und Praktiken.

Das Sprint-Ziel, die für den Sprint ausgewählten PBIs und der Plan zur Umsetzung zusammen werden als Sprint Backlog bezeichnet.

Der Product Owner steht auch im dritten Thema des Sprint Planning beratend zur Seite und ist für Rückfragen erreichbar. Erkennen die Entwickler, dass das Sprint-Ziel nicht realistisch ist, wird gemeinsam neu über das Sprint-Ziel verhandelt. Es können zudem noch Technologieexperten hinzugeholt werden.

16 Mehr Informationen zu den Sprint-Events und deren Dauer im Kapitel „**Scrum Timeboxen**“ auf Seite 65



Das Sprint-Ziel

Das Sprint-Ziel bietet den Entwicklern und dem Product Owner eine gewisse Flexibilität in Bezug auf die zu implementierende Funktionalität. Das Sprint-Ziel ist das Commitment, das zum Sprint Backlog gehört. Die Commitments verbessern die Transparenz und erhöhen des Fokus des Scrum Teams.

Das Sprint-Ziel dient den Entwicklern zur Motivation, gemeinsam während eines Sprints in eine gemeinsame Richtung zu gehen, anstelle in verschiedene Richtungen zu laufen. Dem Product Owner dient es zur Vorbereitung des Kunden und der Stakeholder auf das Increment.

Das Sprint-Ziel kann einerseits vom Product Owner im Sprint Planning vorgeschlagen werden, worauf hin das Scrum Team entsprechende, zum Sprint-Ziel passende, PBIs aus dem Product Backlog auswählt, oder selbstformuliert, oder für einen Vorschlag von PBIs durch den PO gemeinsam ein Sprint-Ziel, welches zu den PBIs passend, ausarbeitet und formuliert.

Das Sprint-Ziel sollte, wenn möglich, immer die Beschreibung einer übergeordneten Funktionalität zusammenhängender

Product Backlog Einträge sein. Ein Sprint-Ziel mit der Aussage alle gewählten PBIs abzuarbeiten ist sowohl für das Team als auch den PO wenig sinnvoll, da hier nicht wirklich ein nutzbares und kundenwertvolles „Ziel“, auf welches hingearbeitet wird, vorliegt. Zudem verliert der Sprint daraufhin seine Kreativität und Flexibilität.

Besser, für das Scrum Team um ein vielfaches wertvoller und aussagekräftiger, sowie wesentlich flexibler und kreativer, ist ein Sprint-Ziel, welches eine Funktionalität oder gar ein Vorhaben beschreiben. PBIs können somit flexibler ausgewählt werden, und der Wert und Nutzen eines Increments somit um ein Vielfaches gesteigert werden.

„Die Suchfunktion sollte um mind. 30% genauer werden“

„Die Applikation sollte zusätzlich zu Oracle auch auf einem SQL-Server lauffähig sein“

„Dem Benutzer soll es möglich sein Zahlungen mit der Kreditkarte zu tätigen“

„Ein minimales Set an Administrationsfeatures sollte verfügbar sein“

„Das Anmelden an der WEB Anwendung und der Datendank unter Einbeziehung von Single Sign on soll möglich sein“



Daily Scrum

Das Daily Scrum Meeting ist ein informelles Synchronisations- und **Planungs-Meeting** von den Entwicklern für die Entwickler und gibt Auskunft darüber was das Scrum Team die vergangenen 24 Stunden erreicht hat, und was es, innerhalb der kommenden 24 Stunden, zu erreichen plant, um das gemeinsame Sprintziel zu erreichen.

Die Planung der Entwickler kann sich dabei an folgenden Leitfragen orientieren, die im Laufe der gemeinsamen Diskussion beantwortet werden:

- Was habe ich seit dem vergangenen Daily Scrum erreicht?
- Was möchte ich bis zum kommenden Daily Scrum erreichen?
- Was hindert mich oder das Scrum Team daran mein Ziel, unsere Ziele und das Sprint-Ziel zu erreichen (Impediments)?

Das Daily Scrum findet, um die Komplexität zu reduzieren, **jeden Tag zur gleichen Zeit am gleichen Ort** statt, und dauert **immer maximal 15 Minuten**.

Detaillierte Diskussionen oder Umplanungen der Arbeit im Sprint

finden häufig offline nach dem offiziellen Daily Scrum Meeting statt, um die Timebox einzuhalten und um zu verhindern, dass Desinteresse während des Meetings aufkommt.

Der Scrum Master sorgt für die Einhaltung der Regel, dass nur die Entwickler des Scrum Teams am Daily Scrum aktiv teilnehmen, sowie dafür, dass das Daily Scrum stattfindet. Die Entwickler sind selbst für die Durchführung des Meetings verantwortlich und zuständig, während der Scrum Master den Entwicklern beibringt, wie sie die Timebox von 15 Minuten einhalten können. Üblicherweise wird während des Daily Scrum Meeting das Taskboard aktualisiert, sowie das Burn Down Chart überprüft und angepasst.

Das Daily Scrum Meeting verbessert die Kommunikation und macht viele andere nicht für Scrum spezifische Meetings überflüssig. Zudem sorgt es für Transparenz und deckt Hindernisse (Impediments) auf, welche das Scrum Team am erfolgreichen Erreichen des Sprint-Ziels hindern könnte.

Das Daily Scrum ist eines der **wichtigsten Meetings** für die Überprüfung und Anpassung in Scrum (**Inspect & Adapt**).



Sprint Review

Das Ergebnis eines Sprints ist immer ein „potenziell auslieferbares wertvolles Increment“.

Das Sprint Review ist ein „Inspect & Adapt“ Meeting bezugnehmend auf das Increment. Das Scrum Team stellt in diesem Review das lauffähige Increment bzw. die Arbeitsergebnisse den Kunden, Endbenutzern bzw. allen Stakeholdern vor, um dies zu „inspizieren“ und dadurch Feedback sowie neue Ideen zu bekommen und somit zu „adaptieren“.

Das Review ist ein informelles Planungs-Meeting, um Anpassungsbedarfe an der Planung und neue Erkenntnisse zu gewinnen (Inspect & Adapt).

Der Product Backlog wird danach angepasst. Neue Prioritäten werden hieraus abgeleitet. Da am Ende eines jeden Sprints ein Review ansteht, ist die Möglichkeit der häufigen und frühzeitigen Korrektur – auch für komplexe Problemstellungen – gegeben.

Das Sprint Review Meeting ist **kein** „Demo“ Meeting. PowerPoint Präsentationen über ein Increment sind genauso unangebracht wie unrealistische Vorführungen über ein Increment. Nur realistische Incremente und die fertige Arbeit des Scrum Teams, wird zur Inspektion vorgeführt, so dass alle Stakeholder das Increment realistisch begutachten, und ggf. auch damit arbeiten können. Schließlich geht es im Sprint Review Meeting darum, ein Increment, welches potentiell dem Kunden ausgeliefert werden kann, zu zeigen.



Sprint Retrospektive

Kaizen

改善

In Scrum und dem empirischen Vorgehen werden das derzeitige Vorgehen als auch äußere Prozesse an sich als unvollkommen akzeptiert. Demzufolge werden in Scrum diese ständig beobachtet und verbessert.

„Inspect & Adapt“ wird jedoch nicht nur auf das Increment selbst, den äußeren Prozessen bzw. dem Vorgehen an sich durchgeführt, sondern auch das gesamte Scrum Team ist stetig dabei, sich zu verbessern. Um dies zu erreichen und um Impediments (Hindernisse) aufzudecken und zu beseitigen, findet die „**Sprint Retrospektive**“ statt. Hierzu wird das gesamte Scrum Team eingeladen.

„Was lief gut?“, „Was ist verbesserungsfähig?“ & „Was werden wir uns im kommenden Sprint vornehmen, besser zu machen?“

Das sind die zentralen Fragen der Retrospektive, wobei die genaue Form variieren kann. Jedes Teammitglied ist aufgefordert, sein Feedback zu liefern.

Der Scrum Master bereitet die Retrospektive anhand verschiedener Arten der Retrospektive¹⁷ vor, moderiert diese auf Grundlage unterschiedlichster Praktiken und sorgt für einen reibungslosen Ablauf des Meetings.

Hier sind vor allem sein Feingefühl, seine Menschenkenntnis, sowie seine Fähigkeit zur Kommunikation und Konfliktlösung gefordert. Das Trennen von Fakten und Emotionen spielt ebenso eine zentrale Rolle.

Die Retrospektive findet immer am Ende jedes Sprints direkt im Anschluss an das Review statt und beendet den Sprint. Wichtig an der Sprint Retrospektive ist, dass Verbesserungspotentiale identifiziert und adressiert werden. Eine Retrospektive ohne konkrete Maßnahmen war nicht erfolgreich.

Teams sollten sich immer darauf einigen, am Ende einer Retrospektive mind. eine Verbesserungsmaßnahme zu identifizieren bzw. ein Impediment im kommenden Sprint zu beseitigen.

¹⁷ Siehe: „**Agile Retrospektives: Making Good Teams Great**“ erschienen 2006 im O'Reilly Vlg. GmbH & Co. ISBN: 978-0977616640

Hört ein Scrum Team auf, die Retrospektive durchzuführen, hört es auf, agil zu sein. Hört ein Scrum Team auf, Scrum zu machen, ist dieses Team aufzulösen.

Die oberste Direktive (Prime Directive) einer Retrospektive ist:

„Unabhängig davon, was wir herausfinden, wir müssen verstehen und glauben, dass in der gegebenen Situation, mit den verfügbaren Ressourcen und mit dem vorhandenen Wissen sowie den individuellen Fähigkeiten eines jeden, jeder sein/ihr Bestes gegeben hat.“¹⁸

Die Retrospektive ist ein disziplinarfreier Raum, eine disziplinarfreie Zone. Niemand wird für seine Transparenz, seinen Mut (Courage) und seine Ehrlichkeit bestraft.

18 Norm Kerth, Project Retrospectives.

Refinement

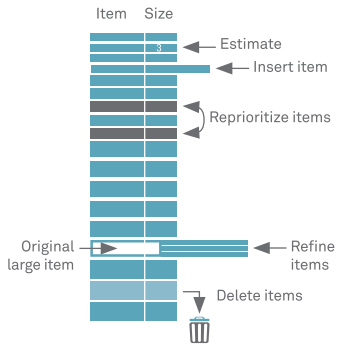
Das „Refinement“ ist eine Good Practice und hilft dem Product Owner und den Entwicklern bei der Erstellung von Planungen und dient zur besseren Vorbereitung auf Sprint Planning Meetings oder auch den Product Backlog zu pflegen.

Ziel ist es, dass der PO weiß, wie „groß“ die PBIs/Stories des Product Backlogs relativ zueinander sind und die Entwickler verstanden haben, was der Product Owner in zukünftigen Sprints oder gar Releases haben möchte. Weiter dient es dem PO als auch den Entwicklern dazu, Klarheit über Backlogeinträge zu erhalten bzw. zu schaffen, und festzustellen, ob die PBIs in ihrer Beschreibung und Art umsetzbar sind.

Zusätzlich kann der PO dadurch mindestens den kommenden Sprint vorbereiten, wenn das Meeting regelmäßig durchgeführt wird, besser sogar das nächste Release planen (rollierend).

19 Wie eingangs bereits erwähnt, werden in diesem Buch auch „Good Practices“ vorgestellt, welche nicht Bestandteil von Scrum sind, sich jedoch aus den Erfahrungen der Scrum Community und der Scrum Coaches bewährt haben und Scrum somit zu besserem Erfolg verhelfen.

Im Refinement schätzen die Entwickler die relative Größe einer jeden, bislang noch nicht geschätzten oder geänderten User Story bzw. Product Backlog Items. Auch EPICs²⁰ können hier „geschätzt“ bzw. erörtert werden.



Je nach Schätzmethode, auf Planning Poker verweisen (Seite ... 51/51) wird hierbei kein Zeitaufwand in Stunden oder Tagen geschätzt, sondern jeder Story wird eine abstrakte Zahl zugeordnet, die in ihrer Größe, relativ zu einer gewählten Referenzstory, entspricht.

Stories, die zu umfangreich sind, werden auf- bzw. heruntergebrochen. Unklare Stories werden diskutiert.

Refinements können regelmäßig oder auch nach Bedarf beliebig oft und zu beliebigen Zeitpunkten durch den PO einberufen werden, und die Timebox hierzu sollte nicht größer als 10% der Sprint-Zeit sein.

Der Product Owner (PO) sollte sich jedoch darüber bewusst sein, dass zu häufige Refinements innerhalb eines Sprints die Gefährdung des Sprint-Ziels zur Folge haben kann und sollte deshalb schon im Planning Meeting ankündigen, wie viele Refinements er in diesem Sprint möchte, sodass die Entwickler dies bei der Aufnahme von User Stories in den Sprint berücksichtigen und die Arbeit, welche im Sprint zu erledigen ist, angepasst werden kann.

Teilnehmer sind der Product Owner, alle Entwickler und auf Wunsch der Scrum Master als Moderator.

Auch Spezialisten können, sofern nötig, am Refinement teilnehmen.

20 EPIC: Ein EPIC kann ein Container für UserStories sein, welcher eine bestimmte Funktion beschreibt. Oft wird ein EPIC auch dafür benutzt, grobgranulare Stories zu beschreiben, welche dann in einzelne UserStories aufgebrochen wird.

Planning Poker

Planning Poker ist eine Good Practice in Scrum, welche ihren Ursprung in XP²¹ hat.

Vorteil des Planning Pokers ist, dass Product Backlog Einträge (Items) mit einer relativen Größe geschätzt werden und nicht der Aufwand.

Jeder Story wird eine abstrakte Zahl zugeordnet, die seiner Größe, relativ zu einer gewählten Referenzstory (wird am Anfang eines Projektes / Produktes identifiziert), entspricht. Der PO stellt seine PBIs vor und beantwortet Fragen der Entwickler. Jede Story wird immer einzeln betrachtet. Jeder Entwickler entscheidet sich für eine Größe und legt diese Karte verdeckt vor sich hin. Wie beim Pokerspiel legt jeder auf Kommando die Karte mit seiner persönlichen Einschätzung in in Story Points, auf den Tisch.

Ist das Ergebnis homogen, gilt die Zahl.

21 XP, auch Extremprogrammierung, ist eine Methode, die das Lösen einer Programmieraufgabe in den Vordergrund der Softwareentwicklung stellt und dabei einem formalisierten Vorgehen geringere Bedeutung zumisst. Diese Vorgehensweise definiert ein Vorgehensmodell der Softwaretechnik, das sich den Anforderungen des Kunden in kleinen Schritten annähert.



Ist das Ergebnis inhomogen (das ist normal), so erklären die Entwickler mit der niedrigsten und mit der höchsten Schätzung, warum sie sich für

diesen Wert entschieden haben. Anschließend wird neu geschätzt. Ist das Ergebnis auch nach insgesamt drei Durchgängen noch inhomogen, so entscheidet der Scrum Master entweder für den häufigsten oder den größten Wert. Ist die Differenz zu hoch und das gesamte Team ist einverstanden, wird diese Story für diesen Sprint nicht herangezogen, so dass der PO mit dem Team in einem Refinement diese erneut ausarbeiten kann.

Entscheidungsgrundlage sollte dabei eine zuvor im Scrum Team getroffene Übereinkunft sein.

Release Planung

Natürlich bekommt der Kunde bzw. die Kunden auch in Scrum eine Aussage darüber, wann welche „Meilensteine“ geplant sind.

Die Release Planung ist ein fester Bestandteil sowohl von klassischer- als auch agiler Vorgehensweisen, in Scrum jedoch, auf Grund der iterativ inkrementellen und empirischen Vorgehensweise anders aufgebaut und wesentlich treffgenauer als z.B. in klassischen Vorgehensmodellen.

Meilensteine, oder Releases, lassen sich an der Anzahl der Sprints ablesen, welche notwendig sind, um die dafür geplanten Features/ EPICs zu implementieren. Voraussetzung hierfür ist eine bekannte Velocity²² des Scrum Teams, sowie ein geschätzter Product Backlog auf deren Grundlage der Product Owner dann zusammenhängende User Stories als Feature oder EPIC identifizieren kann und eine mögliche Aussage über eine mögliche Fertigstellung treffen kann.

Release-Planungen finden regelmäßig statt und basieren auf der Grundlage der gegebenen Velocity des Entwicklungsteams. In

²² Velocity: Durchschnittliche Geschwindigkeit des Scrum Teams pro Sprint gemessen mit z.B. Story Points

einem „Release Planning Meeting“, welches einem „Refinement Meeting“ ähnlich ist, werden gegebene Product Backlog Einträge, welche noch sehr vage formuliert sind, weiter aufgebrochen und geschätzt, mit z.B. Planning Poker. Der Product Owner ist damit dann in der Lage, eine grobe Release Planung vorzunehmen, welche mit Fortschreiten der Sprints immer genauer wird.

Releases können entweder datumsbasiert- oder featurebasiert geplant werden:

Ziel datums-Planung

Das Produkt wird auf ein genaues Datum hin ausgeliefert.

Der PO muss sich die Frage stellen:

Welche Features des Product Backlogs werden zum genannten Datum fertig sein?

Feature-Planung

Das Produkt wird nach Fertigstellung bestimmter Features ausgeliefert.

Der PO muss sich die Frage stellen:

Wann werden die genannten Features fertig sein?

Je nach Ausrichtung einer Release-Planung kann der PO nun eine Aussage treffen entweder:

- Welche neuen Features werden zum genannten Termin fertig sein oder
- wann wird ein neues Release ausgeliefert, auf Grundlage der genannten Features

Eine Kombination aus beiden Vorgehen ist nicht möglich und widerspricht dem empirischen Vorgehen.

Eine stabile und transparente Team Velocity ist Voraussetzung für eine zutreffende Releaseplanung.



Impediment Backlog

Das Pflegen und Führen des bzw. eines sogenannten Impediment Backlogs ist eine Good Practice und wird von Scrum Mastern empfohlen und angewandt, um Impediments des Scrum Teams sichtbar zu machen und zu führen.

Das Führen eines Impediment Backlogs fördert die Transparenz, sowohl des Scrum Masters als auch die des Scrum Teams, da Hindernisse (Impediments) allen transparent und zugänglich sind.

Der **Impediment Backlog** wird **unabhängig** vom **Product Backlog** geführt, muss aber bei den **Sprint Planning Meetings** **berücksichtigt** werden, bezugnehmend auf Verbesserungen, zu denen sich das Team verpflichtet hat, als auch auf Impediments, welche Einfluss auf die Erstellung der Incremente haben.

Der Umgang mit Fehlerkorrekturen und deren Auslieferung, ggf. als Patch, werden in jedem Projekt speziell definiert.

Impediments sind Hindernisse aller Art und nicht unbedingt Bugs.



Die „Definition of Done“ (DoD) – „Fertig!“

Die DoD ist ein Vertrag zwischen den Entwicklern und dem Product Owner und beschreibt exakt, welche Aufgaben erledigt sein müssen und welche Kriterien erfüllt sein müssen, damit & wann etwas fertig ist. Die DoD beschreibt somit die gemeinsame Sprache und Bedeutung von „Fertig“.

Jedes Scrum Team sollte sich auf eine gemeinsame Definition von „Fertig“ (Done) für ein Produkt einigen.

Da die DoD in jeder Retrospektive betrachtet wird ändert sich diese mit zunehmendem Projektverlauf, jedoch aber niemals während eines Sprints.

Veränderungen an der DoD müssen dabei immer Verbesserungen darstellen (verschärfen) - im Nachhinein die Qualität herausdefinieren ist unzulässig (aufweichen).

Es gibt genau eine DoD pro Produkt (und ggf. pro Ebene wie z.B. Task, PBI, Sprint, Release,...), an die sich alle beteiligten Teams halten müssen.

Die DoD ist für alle sichtbar (Transparenz) und auditierbar.

Beispiele für Punkte auf der DoD sind:

- Unit-Test(s) erstellt
- Kompiliert
- Getestet und integriert
- Funktional getestet
- Keine bekannten Fehler
- Code Guidelines eingehalten & Checkstyle fehlerfrei
- Eingecheckt im Master Branch und getaggt
- Integrationstest durchgeführt
- ...

User Stories bzw. Product Backlog Items, welche die DoD nicht erfüllen, kommen zurück in den Product Backlog und werden im nächsten Sprint neu bewertet.



Die „Definition of Ready“ (DoR) – Fertig des POs

Die DoR ist eine Good Practice, welche sich immer weiter etabliert hat und ist ein vom Product Owner und den Entwicklern entwickeltes, geteiltes und somit gemeinsames Verständnis hinsichtlich der Granularität und Qualität eines Product Backlog Items (PBI) welches in einem Sprint Planning Meeting in einen Sprint eingeplant werden soll. Die DoR ist also quasi die DoD des Product Owners für seine Arbeit im Product Backlog (PBL) gegenüber den Entwicklern.

Die „Definition of Ready“ ist somit ähnlich wie die „Definition of Done“, jedoch mit dem Unterschied, dass bei der DoD die Entwickler zuständig sind für die Einhaltung der Kriterien und bei der DoR der Product Owner verantwortlich für die Einhaltung ist.

Die DoR drückt den Anspruch der Entwickler an die Qualität und den Informationsgehalt eines Product Backlog Items aus und der Product Owner sorgt dafür, dass diese Qualität der Backlog Items gewahrt wird.

Die Idee hinter der DoR ist, dass keine unklaren, zu schwammig und lückenhaft verstandenen PBIs in den Sprint eingeplant werden.

Solche PBIs tragen ein hohes Risiko, dass sie nicht innerhalb eines Sprints erledigt werden können.

Schlussendlich geht es um Transparenz und Klarheit und um die Möglichkeit, dass möglichst viele wertvolle Informationen zwischen Entwicklern und Product Owner ausgetauscht werden, und ein gutes Verständnis der Anforderungen herrscht. Jeff Sutherland (Creator von Scrum) sagte einmal: *Fertig ist, wenn das Team sagt: „Ah, wir haben's verstanden“*

Beispiele für Einträge in einer DoR können sein:

- Das Backlog Item (PBI) ist klein genug, um mit mind. 3 weiteren Items in einem Sprint umgesetzt werden zu können
- Das PBI wird von jedem klar verstanden
- Das PBI kann geschätzt werden
- Das Backlog Item hat Akzeptanzkriterien (AKs)
- Die Akzeptanzkriterien sind von jedem Beteiligten klar verstanden
- Alle Abhängigkeiten wurden diskutiert & berücksichtigt
- Die Testdaten sind vorhanden
- UI & UX Entwürfe / Scribbles sind vorhanden

Für agile Entwicklungen in einem Auftraggeber (AG) / Auftragnehmer (AN)-Umfeld hat es sich bewährt, eine gemeinsame DoR zu formulieren, sodass die vom AG gelieferten PBIs einen für den AN annehmbaren und umsetzbaren Stand haben.



Product Backlog

Ausgehend von der gemeinsamen Vision für das Produkt erstellt der Product Owner eine Liste mit Anforderungen für das Projekt bzw. Produkt. Einzelne User Stories beschreiben diese Anforderungen aus Sicht des Kunden und in der Sprache des Kunden. Die Menge der für das Projekt zu erledigenden User Stories ergibt den Product Backlog.

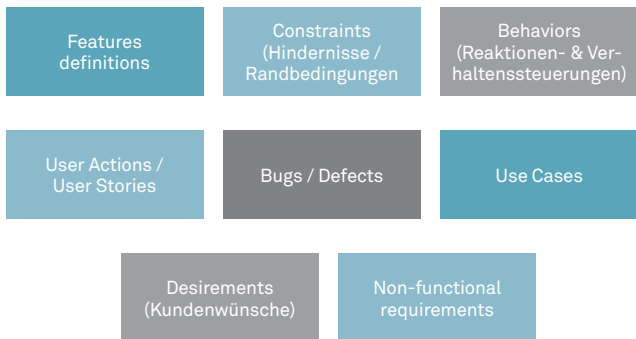
Das Produkt-Ziel ist das Commitment für das Product Backlog. Die Aufgaben im Product Backlog bestimmen, wodurch das Produkt-Ziel erreicht wird.

Auf Grundlage eines gepflegten und geschätzten Product Backlogs kann der Product Owner die kommenden Sprints besser planen, sowie die Release-Planung vornehmen.

Es gibt pro Produkt immer nur **einen Product Backlog**, welcher transparent für alle ist und von jedem jederzeit eingesehen werden kann.

Der Product Backlog enthält alle Arten von Anforderungen und Wünschen für ein Produkt bzw. ein Projekt. Diese Einträge im Product Backlog nennt man **PBIs (Product Backlog Items)**.

Gültige Einträge können unter anderem sein:



Wichtig an dieser Stelle ist die Tatsache, dass auch Bugs & Defects Teile des Product Backlogs sind.

Einige Scrum Master behandeln ShowStopper / High Prio Bugs in einem Kanban Board, welche während eines Sprints abgearbeitet werden können. Gleichzeitig wird sichergestellt, dass das Sprint Ziel nicht gefährdet wird.

Increment

Das Increment stellt einen weiteren konkreten Schritt in Richtung auf das Produkt-Ziel dar. Die Definition of Done ist das Commitment für das Increment. Jedes einzelne Increment baut auf allen vorherigen Increments auf und erweitert diese um den im Sprint hinzugefügten Wert.

Das Increment ist umfassend getestet, so dass sichergestellt wird, dass das Increment nutzbar ist und wertvoll sein kann.

Nur Arbeiten, die die Definition of Done erfüllen, können in ein Increment eingehen.

Scrum Good Practices

Die Erfahrung aus vielen Scrum Projekten zeigt, dass es sich lohnt die „Spielregeln“ des Scrum-Frameworks unbedingt einzuhalten.

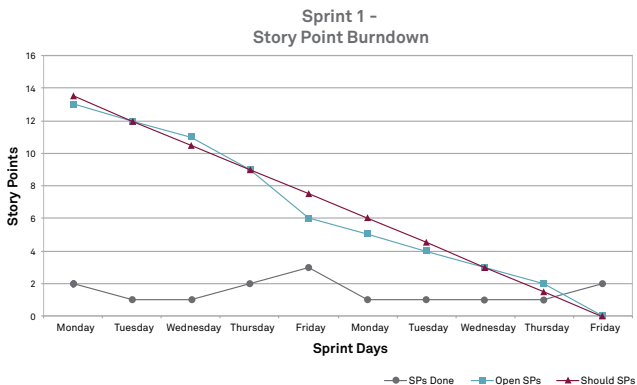
Folgende „Good Practices“ haben sich bewährt:

- Gemeinsam eine klare gemeinsame Vision für das Projekt/ Produkt definieren
- Die beteiligten Menschen stehen im Mittelpunkt, nicht die Technik
- Bereitstellen und Pflegen eines definierten Releaseplanes
- Ein nach Geschäftswert und Risiko geordnetes Product Backlog
- Während des Sprints bleibt das Sprint Backlog stabil
- Das Scrum Team schätzt die Größe der User Stories gemeinsam
- Während des Sprints arbeiten die Entwickler ungestört
- Die ausgelieferte Software ist „fertig“ (DoD)
- Sprint Reviews werten den Projektstand aus
- Sprint Retrospektiven optimieren übergeordnete, organisatorische Prozesse und das Vorgehen
- Projektbezogene Burndown Charts und Metriken erzeugen Transparenz und erlauben das „Fahren auf Sicht“

Sprint Burndown Chart

Das Burndown Chart zeigt klassischerweise die Geschwindigkeit (**Velocity**) des Projektes an und gibt Auskunft darüber, wie viele Story Points ein Team in einem Sprint bereits erledigt hat.

Wichtiger jedoch ist die Darstellung, dass das Burndown Chart eine Aussage darüber liefert, wieviel „**Arbeit**“ das Team noch in diesem Sprint zu „**erledigen**“ hat (verbleibende Story Points).



Die Velocity ist von Scrum Team zu Scrum Team verschieden und kann nicht verglichen werden, da unterschiedliche Scrum Teams unterschiedlich schätzen, Mitglieder unterschiedliche Kompetenzen und Fähigkeiten besitzen sowie unterschiedliche Skills in den Scrum Teams vorhanden sind.

Das Burndown Chart ist vom Scrum Team, für das Scrum Team und hilft zusätzlich dem Product Owner die kommenden Sprints für dieses Scrum Team besser planen zu können.

Das Scrum Team erzeugt damit Transparenz über das, was noch an „**Arbeit**“ in diesem aktuellen Sprint zu erledigen ist, und kann ggfls. frühzeitig reagieren und mit dem Product Owner in eine Planung und Konversation starten.

Das Sprint Burndown Chart wird häufig in Form eines Task Boards gepflegt, auf welchem die Tasks zu den einzelnen User Stories geführt werden. Ist eine Task erledigt und somit eventuell eine UserStory abgearbeitet, kann das BurnDown entsprechend angepasst werden.

Story	To Do		In Progress	To verify	Done
As a user, I... 8 points	Code the ... 9	Test the ... 8	Code the ... DC 4	Test the ... SC 6	Code the ...
	Code the ... 2	Code the ... 8	Test the ... SC 8		Test the ...
	Code the ... 8	Test the ... 4			Test the ...
As a user, I... 5 points	Code the ... 8	Test the ... 8	Code the ... DC 8		Test the ...
	Code the ... 4	Code the ... 6			Test the ... SC 6

Kleines Scrum Vokabular

Nachfolgend eine kurze Auflistung aller Ergebnisverantwortlichkeiten, Artefakte, Ereignisse und Commitments in Scrum:

Ergebnisverantwortlichkeiten

- Product Owner (PO)
- Entwickler
- Scrum Master (SM)

Artefakte

- Product Backlog (PBL)
- Sprint Backlog (SBL)
- Increment

Events

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

Commitments:

- Produkt-Ziel
- Sprint-Ziel
- Definition of Done

Scrum Timeboxen

Event / Sprint	4 Wochen	3 Wochen	2 Wochen	1 Woche
Sprint Planning	8 Stunden	6 Stunden	4 Stunden	2 Stunden
Daily Scrum	15 Minuten			
Sprint Review	4 Stunden	3 Stunden	2 Stunden	1 Stunden
Sprint Retrospektive	3 Stunden	2 Stunden 15 Minuten	1 Stunde 30 Minuten	45 Minuten

Dies ist eine Übersicht aller Timeboxen der jeweiligen Ereignisse in Scrum.

Die Timeboxen für Sprints kürzer als 4 Wochen sind Heuristiken, d.h. Empfehlungen zum Start, um basierend darauf im Scrum Team zur passenden Timebox zu kommen. Scrum erlaubt auch bei Sprints von 1 Woche Dauer z.B. ein Sprint Planning von 8 Stunden.

Software Tools

Meist kommt, wenn das Scrum Team bereits einige Sprints erfolgreich durchgeführt hat oder über mehrere Standorte verteilt arbeitet, der Wunsch nach elektronischer Unterstützung für das Führen des Product Backlogs und Sprint Backlogs, sowie das automatische Generieren von Burndown Charts, auf.

Heutige Werkzeuge decken dies für agile Vorgehensweisen und Scrum sehr gut ab und unterstützen zusätzliche Funktionen für alle Phasen des Application- & Product Lifecycle.

Glossar

Nachfolgend ein Glossar wichtiger Begriffe im Scrum Umfeld.

Die Liste bzw. Tabelle erhebt keinen Anspruch auf Vollständigkeit.

Begriff	Beschreibung
Backlog	siehe Product Backlog (PBL) und Sprint Backlog (SBL)
Burndown Chart	Ein Diagramm, welches die Entwicklung des (Rest-)Aufwandes bzw. des verbleibenden Arbeitspensums im Verhältnis zur verbleibenden Zeit zeigt. Sie können optional angewandt werden, um die Fortschritte transparent darzustellen.
Daily Scrum	Tägliche, auf 15 Minuten begrenzte, Synchronisation der Entwickler, um den nächsten Arbeitstag im Detail hinsichtlich des Sprint-Ziels zu planen. Aktualisierungen werden in das Sprint-Ziels eingepflegt.
Definition of Done (DoD)	Ein gemeinsames Verständnis hinsichtlich der Erwartungen, die Software (und andere Produktentwicklungen) erfüllen muss, um in den Produktionsprozess einfließen zu können.
Definition of Ready (DoR)	Von Product Owner und Entwicklern gemeinsam entwickeltes und geteiltes Verständnis hinsichtlich der Granularität und Qualität eines Product Backlog Items (PBI), welches im Sprint Planning in den Sprint geplant werden soll.
Entwickler	Alle Mitglieder des Scrum Teams, die am Increment mitarbeiten, unabhängig technischer, funktionaler oder anderer Spezialisierungen.
EPIC	Eine Anforderung auf geringer Granularitätsstufe welche aufgrund ihrer geringen Tiefe bzw. ihrer Größe nicht in einem Sprint entwickelt werden kann sondern sich über mehrere Sprints erstreckt und deswegen genauer analysiert und ggf. auf Sprintgröße geschnitten werden muss. Ein Epic dient auch als Container hinsichtlich seines Inhaltes für mehrere UserStories.

Begriff	Beschreibung
Feature	Ein Feature beschreibt eine gekapselte Funktionalität / Eigenschaft eines Systems oder Produktes welches ein oder mehrere Bedürfnisse der Nutzer befriedigt.
Increment	Ein Arbeitsergebnis / Produktanteil, welches den zuvor erstellten „Produktinkrementen“ neue Funktionalität / Mehrwert hinzufügt. Die Produktinkremente formen als Ganzes (Summe) das angestrebte Produkt. Ein Inkrement ist in der Regel das Ergebnis eines Sprints.
Inspect & Adapt	Zwei der Scrum Grundfesten (Säulen) – üblicherweise zusammen genannt. Sind Basis für die Retrospektive. Das Prinzip „Inspect & Adapt“ führt zu einem empirischen Vorgehen.
Iteration	Beschreibt allgemein einen Prozess mehrfachen Wiederholens gleicher oder ähnlicher Handlungen zur Annäherung an eine Lösung oder ein bestimmtes Ziel.
Planning Poker	Eine Technik um beim agilen Schätzen die verschiedenen Annahmen herauszufinden und ein vorzeitiges „Anchoring“ zu vermeiden.
Product Backlog (PBL)	Eine jederzeit geordnete Liste, die umfassend alle Informationen inklusive der angestrebten Funktionalitäten, Anforderungen, und Rahmenbedingungen beinhaltet. Der Product Owner ist der Manager des Product Backlogs.
Product Backlog Item (PBI)	Ein Eintrag im Product Backlog. Kann jegliche Form von Anforderung sein.
Product Backlog Refinement	Die Aktivität, in welcher der Product Owner gemeinsam mit den Entwicklern das Product Backlog hinsichtlich der Anforderungen für das nächste Planning inhaltlich und qualitativ aufbereitet.
Product Owner (PO)	Die Ergebnisverantwortlichkeit in Scrum, deren Aufgabe darin besteht, den Wert eines Produktes zu maximieren. Der Product Owner verantwortet die wirtschaftlichen und funktionalen Anforderungen des zu erstellenden Produktes gegenüber den Entwicklern und den Stakeholdern..
Produkt-Ziel	Das Produkt-Ziel ist das langfristige Ziel, gegen welches das Scrum Team plant. Es gibt immer nur ein Produkt-Ziel

Agile- / Scrum Abkürzung	Beschreibung
Ready	Siehe Definition of Ready (DoR)
Refinement Meeting	Siehe Product Backlog Refinement
Release	Fasst mehrere Sprints zu einer größeren (Liefer-) Einheit (oft drei Monate) zusammen.
Scrum	Ein organisatorisches Management Rahmenwerk, welches Teams in komplexen Produktentwicklungen unterstützen soll. Scrum basiert auf den Scrum Teams mit ihren verbundenen Ergebnisverantwortlichkeiten, Events, Artefakten, Commitments und Regeln (definiert im Scrum Guide).
Scrum Guide	Die Spielregeln für Scrum, verfasst und veröffentlicht von Ken Schwaber und Jeff Sutherland (die beiden Väter von Scrum). Der Scrum Guide besteht aus der Beschreibung von Ergebnisverantwortlichkeiten, Events, Artefakten, Commitments und der Regeln, durch die sie miteinander verbunden sind.
Scrum Master (SM)	Die Ergebnisverantwortlichkeit im Scrum Team, die für das Coaching, die Vermittlung und die Unterstützung des Scrum Teams und seiner Umwelt verantwortlich ist. Seine Aufgabe besteht darin, dass Verständnis für und die Nutzung von Scrum allen am Produktentwicklungsprozess Beteiligten zu vermitteln und Lösungswege für etwaige Fragen im Scrumprozess selbst (nicht fachlich auf das Produkt bezogen) zu finden bzw. zu begleiten.
Scrum Team	Ein sich selbst-organisierendes Team, bestehend aus einem Product Owner, Entwicklern und einem Scrum Master.

Agile- / Scrum Abkürzung	Beschreibung
Selbstorganisation	Selbstorganisation als Managementprinzip, welches den Scrum Teams ermöglicht, autonom ihre Arbeit zu organisieren. Selbstorganisation erfolgt innerhalb bestimmter Grenzen und vorgegebener Ziele. Die Teams entscheiden selbst, wie sie ihre Arbeit am besten gestalten, anstatt nur detaillierte Vorgaben von außen zu befolgen.
Sprint	Zeitlich beschränktes Ereignis von bis zu 4 Wochen (oder 30 Tagen / einem Kalendermonat), je nach Scrum-Team und Produkt auch kürzer. Dient als Rahmen für alle anderen Scrum Ereignisse und Aktivitäten. Sprints werden nacheinander ohne organisatorische Zwischenschritte oder zeitliche Zwischenräume durchgeführt.
Sprint Backlog (SBL)	Eine Übersicht aller Entwicklungsarbeit, die für das Erreichen des Sprintziels erfolgen muss. Das Sprint Backlog wird durch die Entwickler erstellt, organisiert und abgearbeitet.
Sprint-Ziel	Eine knappe Beschreibung des zu erreichenden, übergeordneten Zieles eines Sprints welches sich oft auf eine wirtschaftliche oder unternehmerische Anforderung bezieht. Das Sprint-Ziel ist unveränderlich und kann dazu führen, dass geplante Funktionalität im Sprint zugunsten der Erreichung des Sprint-Ziels überarbeitet wird.
Sprint Planning	Ein Ereignis von maximal 8 Stunden. Dieses Ereignis hat das Ziel, dass das Scrum Team die wertvollsten und wichtigsten Aufgaben aus dem Product Backlog auswählt und in das Sprint Backlog überträgt.
Sprint Retrospektive	Dieses Ereignis von maximal 3 Stunden. Dieses Ereignis dient dem Scrum Team dazu, den vergangenen Sprint zu evaluieren und Verbesserungen im Scrum- Entwicklungsprozess zu planen, die im nächsten Sprint umgesetzt werden.
Sprint Review	Ein Ereignis von maximal 4 Stunden. Dieses Ereignis dient dem Scrum Team und den Stakeholdern dazu, das im Sprint entstandene Produktinkrement zu bewerten, den Fortschritt im Gesamtprozess zu evaluieren und das Product Backlog für die kommenden Sprints mit dem eingeholten Feedback anzupassen.

Begriff	Beschreibung
Stakeholder	Eine Person/Interessensgruppe außerhalb des Scrum-Teams mit einem spezifischen Interesse an dem zu entwickelnden Produkt. Stakeholder können vom Product Owner vertreten werden und sind als Feedbackgeber aktiv in das Sprint Review eingebunden.
Story Point (SP)	Story Points sind eine künstliche Maßeinheit für die Größe einer User Story (Anforderung). Sie umfassen eine Bewertung von Aufwand, Komplexität und Risiko und sind daher weit mehr als eine reine Aufwandsschätzung.
Transparenz	Einer der drei Eckpfeiler von Scrum. Transparenz ist die notwendige Voraussetzung für die Inspektion und Adaption und die Basis, um sowohl das Produkt, als auch das Entwicklungsvorgehen, die Interaktionen zwischen den Individuen innerhalb und außerhalb des Scrum Teams zu verbessern.
Timebox	Maximale Menge an Zeit die man einer Arbeit oder einem Meeting zugesteht.
User Stories	Optionale Form, Anforderungen herauszuarbeiten und zu beschreiben, die sich im Zusammenspiel mit agiler Entwicklung bewährt hat. Benutzt wird die Form: „Als <Rolle> möchte ich <Funktion> damit ich <Wertschöpfungsgrund>“
Velocity	Eine optionale, allerdings oft genutzte Angabe der durchschnittlichen Menge an Product Backlog Items Elementen (oder Story Points) welche in ein Produktinkrement während eines Sprints eingeflossen sind (also umgesetzt wurden). Die Velocity hilft primär dem Product Owner dabei, eine Release-Planung zu erstellen, d.h. ein mögliches Datum für die Fertigstellung eines bestimmten Funktionsumfang abzuschätzen.

Kontakt

XT - msg Research

research@msg.group

msg.direct/msg-agility-de



Urheberrechtshinweis

Alle Inhalte dieses Buches, insbesondere Texte, Fotografien und Grafiken, sind urheberrechtlich geschützt (Copyright). Das Urheberrecht liegt, soweit nicht ausdrücklich anders gekennzeichnet, bei Michael Springer sowie der msg systems ag. Bitte fragen Sie uns (Die Kontaktdaten finden Sie oben auf dieser Seite), falls Sie die Inhalte dieses Buchs verwenden möchten.

Inhalte die unter der CreativeCommons-Lizenz veröffentlicht und speziell gekennzeichnet wurden, dürfen nach den maßgeblichen Lizenzbedingungen verwendet werden.

Wer gegen das Urheberrecht verstößt (z.B. die Inhalte unerlaubt auf die eigene Homepage kopiert oder in eigenen Publikationen verwendet), macht sich gem. § 106 ff Urhebergesetz strafbar.

Er wird zudem kostenpflichtig abgemahnt und muss Schadensersatz leisten. Kopien von Inhalten oder des gesamten Werkes können im Internet ohne großen Aufwand verfolgt werden.

© 2015-2022 / msg Research / research@msg.group / msg systems ag

Go Scrumming – Scrum Tools für die Hosentasche

2. Auflage / Version 2.0

2022

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München

Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113

www.msg.group | info@msg.group