

# SCRUM



## Schnellstart-Guide

Agiles Projektmanagement und Scrum  
schnell lernen und erfolgreich anwenden



Abgestimmt mit *Scrum Guide*™ 2020

**ROLAND WANNER**

# SCRUM

---



**Schnellstart-Guide**

**Agiles Projektmanagement  
und Scrum schnell lernen und  
erfolgreich anwenden**

**ROLAND WANNER**

## **Kontakt zum Autor:**

Roland Wanner

E-Mail: [info@rolandwanner.ch](mailto:info@rolandwanner.ch)

Internet: [www.rolandwanner.ch/](http://www.rolandwanner.ch/)



## **Herstellung:**

Amazon Media EU S.à r.l, Luxembourg

## **Haftungsausschluss**

Dieses Buch enthält Informationen über Scrum und agiles Projektmanagement. Es wurde zu Informations- und zu Weiterbildungszwecken geschrieben. Für den professionellen Einsatz empfiehlt sich die Unterstützung durch eine kompetente Fachperson.

Trotz größter Sorgfalt, dieses Buch so vollständig und korrekt wie möglich zu machen, ist nicht auszuschließen, dass es Fehler enthält, typografische oder inhaltliche. Deshalb ist dieser Text nur als genereller Leitfaden und nicht als alleinige Informationsquelle über Scrum und agiles Projektmanagement zu verwenden.

Der Autor, Herausgeber und die zitierten Quellen haften nicht für etwaige Verluste, die aufgrund der direkten oder indirekten Umsetzung der in diesem Buch verwendeten Beschreibungen entstehen könnten.

Themen in diesem Buch sind: Scrum, agiles Projektmanagement, IT und Software Projekte, Sprint, Timeboxing, Product Owner, Scrum Master, Daily Scrum und weitere.

Bei Fragen oder Anregungen kontaktieren Sie bitte:

[info@rolandwanner.ch](mailto:info@rolandwanner.ch)

Alle Rechte, einschliesslich derjenigen des auszugsweisen Abdruckes sowie der fotomechanischen und elektronischen Wiedergabe, vorbehalten.

Es wird darauf hingewiesen, dass die im Buch verwendeten Software- und Hardwarebezeichnungen sowie Markennamen und Produktbezeichnungen, wie z. B. The Scrum Guide™, der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Copyright © 2021 Roland Wanner

ISBN: 978-1699306468

1. Auflage Januar 2021

## **Bibliografische Information der Deutschen Bibliothek:**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über

<http://dnb.ddb.de> abrufbar.

# Inhalt

Vorwort.....	7
<b>1 Agiles Projektmanagement</b>	<b>11</b>
Agiles Projektmanagement im Überblick.....	12
Der Unterschied zwischen klassischen und agilen Projekten.....	14
Das Agile Manifest.....	17
Agile Prinzipien.....	18
Agile Methoden.....	20
Die Struktur des agilen Projektmanagements.....	23
<b>2 Das Scrum Framework</b>	<b>25</b>
So entstand Scrum.....	25
Die drei Säulen von Scrum.....	28
Scrum im Schnellüberblick.....	31
<b>3 Das Scrum-Team</b>	<b>35</b>
Der Product Owner.....	39
Die Developer.....	42
Der Scrum Master.....	43
Die Aufgaben des Scrum Masters.....	43
Weitere Rollen.....	46
Das Scrum-Team und sein Umfeld.....	47
<b>4 Die Scrum-Werte</b>	<b>49</b>
Die Scrum-Werte als das Fundament.....	50
Verpflichtung (Commitment).....	50
Mut (Courage).....	51
Fokus (Focus).....	51
Offenheit (Openness).....	52
Respekt.....	52
Die Scrum Werte leben.....	52

<b>5</b>	<b>Die Scrum Events</b>	<b>53</b>
	Der Sprint .....	55
	Das Sprint Planning.....	58
	Die Planungsschritte.....	58
	Das Daily Scrum .....	62
	Das Daily Scrum vorbereiten und durchführen .....	62
	Das Sprint Review .....	66
	Vorbereitung und Ablauf des Sprint- Reviews.....	67
	Die Sprint-Retrospektive .....	70
	Vorbereiten der Retrospektive .....	71
	Die Retrospektive durchführen .....	72
<b>6</b>	<b>Die Scrum-Artefakte</b>	<b>75</b>
	Das Product Backlog .....	77
	Übersicht über das Product Backlog .....	79
	Das Product Backlog priorisieren .....	81
	Priorisierung nach MoSCoW.....	82
	Priorisieren ist eine wiederkehrende Optimierungsaufgabe ....	83
	Das Sprint Backlog .....	84
	Das Sprint Backlog auf dem Task Board .....	85
	Das Increment .....	87
<b>7</b>	<b>Das Anforderungsmanagement</b>	<b>91</b>
	Produktvision, Produktziel, Sprint-Ziel .....	92
	Die Produktvision.....	93
	Anforderungsmanagement in Scrum .....	95
	Der Anforderungsworkshop .....	96
	Die Merkmale guter Anforderungen .....	98
	User Stories .....	99
	Nichtfunktionale Anforderungen erfassen .....	100
	Unterschied zwischen Epic, Theme, User Story und Task.....	102
<b>8</b>	<b>Planung in Scrum</b>	<b>105</b>
	Wann startet das Projekt? .....	106
	Bevor das Projekt startet .....	106

Releaseplanung.....	108
Aufwandschätzung .....	112
Aufwandschätzung auf zwei Ebenen.....	112
Das Estimation Meeting .....	114
Mit Punkten schätzen .....	115
User Stories sollten „klein“ sein.....	116
Die optimale Sprintdauer.....	116
Die Entwicklungsgeschwindigkeit bestimmen.....	117
<b>9 Den Sprint durchführen</b>	<b>119</b>
Artefakte und Events im Sprintablauf.....	119
Die Arbeit ausführen.....	121
Den Sprint-Umfang anpassen .....	122
Einen Sprint abbrechen .....	122
Den Fortschritt überwachen.....	123
So wird das agile Projekt gesteuert.....	127
Der Scrum Workflow.....	128
<b>10 Glossar</b>	<b>131</b>
<b>11 Anhang</b>	<b>141</b>
Internet-Links und empfohlene Literatur .....	141
Über den Autor .....	143
Literaturverzeichnis .....	144
Stichwortverzeichnis.....	146



Dieses Buch basiert auf dem **Scrum Guide™ 2020**  
von Ken Schwaber und Jeff Sutherland

## Vorwort

---

Ich gratuliere Ihnen, dass Sie sich mit agilem Projektmanagement und Scrum beschäftigen wollen! Dies ist das ideale Buch, mit dem Sie schnell alles Wichtige über agiles Projektmanagement und Scrum lernen oder bestehendes Wissen weiter vertiefen. Dieses Buch ist ein praxisbezogener Leitfaden, welcher Sie durch alle Stationen eines Scrum-Projektes führt.

Scrum und Agile sind seit ein paar Jahren eines der wichtigsten Managementthemen und auch agiles Projektmanagement wird in Zukunft stark an Bedeutung gewinnen – nicht nur für Software-Entwicklungsprojekte. Mit dem Wissen in diesem Buch sind Sie auf dem neusten Stand und für eine interessante Zukunft gut gerüstet.

In diesem Buch erhalten Sie zum Start eine kurze und allgemeine Einführung in das agile Projektmanagement. Diese legt dann die Basis für den größten Teil des Buches, dass agile Projektmanagement mit Scrum für Software-Entwicklungsprojekte.

### **Schneller, günstiger, besser und mehr Business Value**

Agile Methoden sind auf dem Vormarsch, nicht nur im Projektmanagement. Auch in anderen Managementbereichen versucht man langsam den Nutzen daraus zu ziehen, Prozesse zu verschlanken, sich noch mehr auf den Kunden und den Kundennutzen zu fokussieren, noch schneller auf Marktveränderungen zu reagieren und mehr mit selbstorganisierenden Teams zu „experimentieren“. Agile Methoden haben aber auch das Ziel, die Arbeit sinnvoller, interessanter und lohnenswerter zu gestalten. Wenn Sie schon etwas älter sind, dann kommen Ihnen einige dieser Themen vielleicht bekannt vor. Selbstorganisierende Teams wurden zum Beispiel schon seit den 1960er Jahren in verschiedenen Unternehmen erfolgreich eingesetzt – haben sich aber bis nie richtig etabliert.



### **Erfolgreiche Managementmethoden wiederentdeckt**

Die Prinzipien, Methoden und Werte, auf denen das agile Projektmanagement aufbaut, gibt es schon seit Jahrzehnten. Sie stammen praktisch alle vom Toyota Production System.

Das Toyota Production System (TPS), welches nach dem zweiten Weltkrieg entstand war ein Treiber für viele wirkungsvolle Managementmethoden, wie Lean Production, Lean Management, Business Process Reengineering und Six Sigma. Auch selbstorganisierende Teams waren ein Kernelement des TPS und wurden dann in den 1950er und 1960er in verschiedenen japanischen und später in amerikanischen Unternehmen eingesetzt.

Alle diese Managementmethoden waren in den betreffenden Unternehmen sehr erfolgreich, haben sich aber erstaunlicherweise in der Wirtschaft nie richtig durchgesetzt und gingen fast vergessen. Einzig die Autoindustrie hat vom Toyota Production System und Lean Production gelernt und davon stark profitiert.

Die Arbeit in Projekten in der Industrie, auf dem Bau und in anderen Branchen wird schon seit Jahrhunderten in sequenziellen, aufeinander aufbauen Phasen ausgeführt, die. Als die Informationstechnologie und Softwareentwicklung in den 1960er Jahren entstand wurde die sequenzielle Vorgehensweise auch in diese Projekte übernommen. In den Folgejahren wurden verschiedene mehr oder weniger erfolgreiche iterative Softwareentwicklungs-Methoden entwickelt, um die Softwareentwicklung effizienter zu machen und um besser auf verändernde Marktanforderungen reagieren zu können

1995 präsentierten Jeff Sutherland und Ken Schwaber zusammen ein Dokument, welches die Scrum Methode für Softwareprojekte beschrieb. Sie entwickelten, basierend aus den sehr erfolgreichen, aber schon fast vergessenen Werten, Prinzipien und Methoden, die Sie vorhin kurz kennengelernt haben, die erfolgreichste Projektmanagement-Methode für die Softwareentwicklung – seit Software entwickelt wird – und dies ist SCRUM.

## **Für wen wurde dieses Buch geschrieben?**

Dies ist ein Buch für alle, die am agilem Projektmanagement interessiert sind und wissen wollen, wie die bekannteste agile Methode in der Softwareentwicklung, Scrum, funktioniert.

Dies ist kein Buch mit allen Details, vielen Beispielen, Geschichten und Ausschweifungen. Für das gibt es umfassendere und viel teure Bücher. Hier lernen Sie konzentriert das Wichtigste über agiles Projektmanagement und Scrum.

Ob Sie bereits im Softwarebereich arbeiten, Student, Projektauftraggeber für Software oder bereits in einem agilen Projekt arbeiten – dieses Buch liefert Ihnen in kompakter Form das notwendige Wissen, agiles Projektmanagement und besonders Scrum besser zu verstehen und erfolgreich anzuwenden.

## **So ist das Buch aufgebaut**

Zuerst lernen Sie, wie das agile Projektmanagement entstanden ist und was der Unterschied ist zum „normalen“ Projektmanagement. Dann lesen Sie, was das „agile Manifest“ ist und was die agilen Prinzipien für eine Bedeutung für Scrum und die anderen agilen Methoden hat.

Der Hauptteil dieses Buches beschäftigt sich natürlich mit Scrum. Zuerst gebe ich Ihnen einen **schnellen Überblick über das Scrum Framework**. Damit erhalten Sie innerhalb von fünf Minuten ein Grundwissen und verstehen so später die Zusammenhänge besser.

Die folgenden Kapitel geben Ihnen einen Überblick über die Werte von Scrum, die Scrum Events und die Artefakte.

## **Ein ideales Nachschlagewerk**

Wenn Sie dieses Buch gelesen haben, dann ist es für Sie später auch ein ideales Nachschlagewerk. Am Schluss des Buches finden Sie ein umfangreiches **Glossar** und ein hilfreiches **Stichwortverzeichnis**, mit dem Sie schnell ein bestimmtes Thema im Buch finden und sich darüber informieren können.

## Vorwort

Nicht zuletzt eignet sich dieses Buch ausgezeichnet für **Schulungen** und **das Studium** im Bereich Softwareentwicklung.

### Der Scrum Guide

„The Scrum Guide™“ von Ken Schwaber und Jeff Sutherland ist die offizielle Guideline für Scrum. Er definiert die Grundlagen von Scrum und wird periodisch aktualisiert. Dieses Buch basiert auf der neusten Version des **Scrum Guides vom November 2020**.

Dieses Buch gibt Ihnen einen Einblick in das agile Projektmanagement und Scrum mit vielen hilfreichen Abbildung und geht weit über den Scrum-Guide hinaus.

### Terminologie in diesem Buch

Ich verwende in diesem Buch bewusst die bekannten, englischsprachigen Scrum-Begriffe für die *Rollen*, *Artefakte* und *Events*. So werden Sie nicht mit neuen Wortschöpfungen unnötigerweise verwirrt.

Dieses Buch ist absichtlich in der männlichen Form geschrieben, natürlich nicht um das weibliche Geschlecht auszugrenzen oder zu diskriminieren, sondern damit der Text einfacher zu lesen ist. Ich hoffe die weiblichen Leser haben Verständnis dafür.

# 1

# Agiles Projektmanagement

**A**gile Methoden werden in immer mehr Projekten eingesetzt – in der Softwareproduktentwicklung schon seit vielen Jahren erfolgreich, bei anderen Projektarten stehen wir hier noch am Anfang. Der Trend zeigt aber eindeutig, immer mehr Unternehmen beschäftigen sich mit agilen Methoden oder sind bereits an deren Einführung. Auch in traditionellen Industrien, wie zum Beispiel im Automobil- oder Flugzeugbau, führt man damit Projekte bereits erfolgreich durch. Beispiele dafür sind Toyota, BMW oder SAAB Technologies.

Agile Methoden wie Scrum zu lernen und zu verstehen ist relativ einfach. Die agilen Werte und Grundprinzipien zu verinnerlichen und zu leben ist hingegen einiges schwieriger – und hier haben viele Unternehmen noch Mühe.

Der Erfolg agiler Methoden auf Unternehmensebene kommt schlussendlich aber nur zustande durch radikale Veränderungen innerhalb der Organisationen und hier sind wir noch weit davon entfernt große Fortschritte zu machen. Auf Projektebene sind wir hier hingegen schon sehr weit fortgeschritten.

Agile Methoden haben einen radikalen Einfluss auf die Führungs- und Kompensationssysteme in Unternehmen. Manager verlieren auch Macht und Einfluss bei selbstorganisierenden Teams. Dies wird die Veränderung, speziell in der Unternehmenskultur nicht einfach machen.

In der Softwareentwicklung werden agile Methoden und Scrum schon seit mehr als 20 Jahren erfolgreich angewendet. Ein erster, wichtiger Schritt ist also schon gemacht!

## Agiles Projektmanagement im Überblick

---

Imposante Projekte wurden schon vor tausenden von Jahren durchgeführt. Denken Sie z. B. an die Steinstruktur von Stonehenge, die 3500 Jahre v. Chr. und die ägyptischen Pyramiden, die 2500 v. Chr. erbaut wurden oder in der neueren Zeit an die mittelalterlichen Burgen, Festung, Schlösser und Kathedralen, die Dampfmaschine, Autos, die Atombombe oder die Wolkenkratzer. Dies waren teilweise riesige und komplexe Projekte für Ihre Zeit.

Softwareentwicklungsmethoden gab es bis in die 1960er Jahre noch keine. Der Systems Development Life Cycle (SDLC) war der erste, der in dieser Zeit entstand, mit dem Ziel, große, funktionale Business Systeme zu entwickeln. Alle Projekte wurden bis in die 90er Jahre nach dem sequenziellen Wasserfallmodell abgewickelt (siehe Abbildung Seite 14). Das heißt, alle Anforderungen wurden zu Projektbeginn festgelegt, dann wurden Konzepte, Spezifikationen und Pläne erstellt und dann das Produkt gebaut und im Markt eingeführt.

Die Softwareentwicklung in den 1990er Jahren wurde durch die objektorientierte Programmierung, durch den Aufstieg des Internets und den [Dot.com-Boom](#) geprägt. Hier war Time-to-Market und Firmenwachstum entscheidend, d.h. die Entwicklungszyklen für Software wurden immer kürzer.

### **Das Wasserfall-Modell war nicht mehr geeignet**

Mit dem starren Wasserfall-Modell war man in der Software-Entwicklung deshalb immer weniger zufrieden, besonders weil die Projekte immer komplexer wurden, Produktlebenszyklen immer kürzer und das Umfeld und die Anforderungen dynamischer. Man benötigte immer schneller brauchbare Software, nicht mit allen Funktionen, aber mit den Wichtigsten. Leichtgewichtiger, flexibler und schneller sollte die Softwareentwicklung werden und weniger Administration waren gewünscht.

Neue Methoden sollten den Softwareentwicklungsprozess flexibler und schlanker gestalten – als Gegenbewegung zu den eher schwergewichtigen und bürokratischen, traditionellen Methoden, wie eben z.B. dem Wasserfall-Modell. Diese Forderungen stießen eine aktive Entwicklung von verschiedenen Methoden in der Softwareentwicklung an wie z. B: Rapid Application Development (RAD), Rational Unified Process (RUP) oder Extreme Programming (XP).

Das Interesse an Extreme Programming ebnete den Weg auch für andere agile Prozesse und Methoden. Die Bezeichnung „agil“ für diese Art der Softwareentwicklung wurde ausgewählt von 17 Vertretern von verschiedenen Softwareentwicklungsmethoden im Februar 2001, bei einem Treffen in Utah (USA). Dies als Ersatz für das bis dahin gebräuchliche leichtgewichtig (engl. lightweight). Bei diesem Treffen wurde auch das Agile Manifest (siehe Seite 17) formuliert. Daraus hat sich dann im Laufe der Jahre die Bezeichnung „agiles Projektmanagement“ entwickelt, denn nicht nur Softwareprojekte lassen sich agil planen, umsetzen und steuern, sondern auch andere Projektarten.

Das Ziel agiler Softwareentwicklung ist es, den Entwicklungsprozess flexibler und schlanker zu machen, als dies bei den klassischen Vorgehensmodellen der Fall ist. Agile Softwareentwicklung zeichnet sich durch selbstorganisierende Teams, sowie eine iterative und inkrementelle Vorgehensweise aus. Es wird versucht mit geringem bürokratischem Aufwand und weniger Regeln auszukommen, damit man sich schnell an Veränderungen anpassen kann, ohne dabei das Risiko für Fehler zu erhöhen.

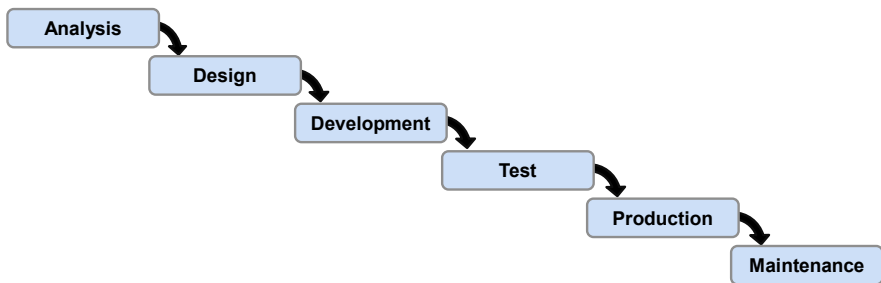
# Der Unterschied zwischen klassischen und agilen Projekten

---

Bei **klassischen Projekten** werden vom internen oder externen Kunden (Auftraggeber) zu Projektbeginn klare Ziele und Anforderungen definiert, die sich während der Projektdurchführung möglichst nicht mehr ändern. Am Ende des Projektes erhält der Kunde das, was er am Anfang definiert hat.

Das Projekt wird strikt in nacheinander folgenden Phasen durchgeführt. Eine vorhergehende Phase muss beendet sein, bevor mit der nächsten gestartet werden kann. Das Projektresultat entsteht im Ablauf der Phasen, bis es dann am Ende der letzten Phase vollständig fertiggestellt ist. Dieser Ablauf wird Wasserfall-Modell genannt.

Je weiter das Projekt fortschreitet, desto weniger kann der Kunde hier Einfluss auf das Endergebnis nehmen. Eine große Einschränkung beim Wasserfallmodell ist, dass jede Änderung oder neue Anforderungen, die der Kunde in einer späteren Projektphase noch umgesetzt haben will, ein Mehrfaches kostet, als wenn er diese am Anfang definiert hätte.



*Abbildung 1: Das Wasserfall-Modell*

Sie können sich vermutlich gut vorstellen, wie man ein Haus baut. Wenn bei diesem Projekt die Bauarbeiter bereits die Wände mauern und Sie wünschen zu diesem Zeitpunkt noch ein zusätzliches Zimmer, dann wird dies sehr teuer oder es ist fast unmöglich.

**Agile Methoden**, z. B. Scrum werden in Entwicklungsprojekten (besonders von Software) verwendet, um sich der hohen Dynamik der Ziele, Anforderungen, Umfeld und Erwartungen anzupassen. Man setzt dabei z. B.:

- auf enge Zusammenarbeit zwischen dem Kunden, dem Product Owner und dem sich weitgehend selbst organisierenden Team
- auf kurze Entwicklungszyklen, nach denen Änderungen und neue Anforderungen in die Planung zusätzlich aufgenommen werden können.

Damit macht es das agile Projektmanagement dem Auftraggeber oder den Stakeholdern möglich, neue Anforderungen während dem gesamten Projektablauf einzubringen oder bestehende Anforderungen zu ändern. So kann man auf kurzfristige Marktbedürfnisse flexibel einzugehen – und dies, ohne ein exponentielles Kostenwachstum zu verursachen. Dabei muss natürlich das Gesamtbudget trotzdem im Auge behalten werden.

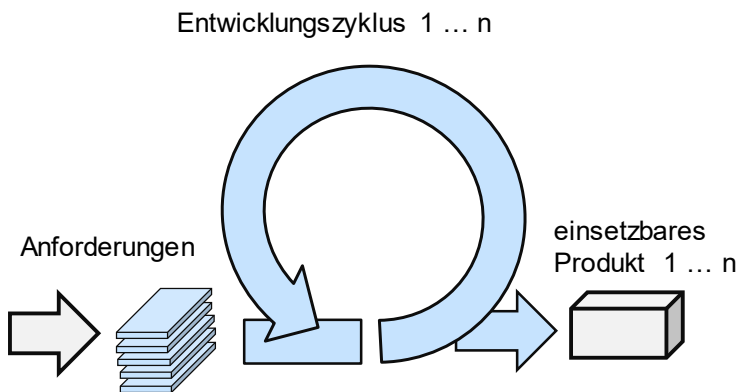


Abbildung 2: Das agile Vorgehen



# 6

## Die Scrum-Artefakte

**N**eben den Scrum-Team Mitgliedern und den fünf Events definiert Scrum auch drei Artefakte. Diese haben den Zweck, die Transparenz von Schlüsselinformationen zu maximieren. So haben alle, die sie überprüfen, die gleiche Grundlage für Anpassungen. Die Artefakte repräsentieren Wert (Business Value) oder Arbeit.

**Jedes Artefakt enthält ein Commitment**, um sicherzustellen, dass Informationen bereitgestellt werden, welche Transparenz und Fokus verbessern, an denen der Fortschritt gemessen werden kann:

- Für das **Product Backlog** ist es das Produkt-Ziel
- Für das **Sprint Backlog** ist es das Sprint-Ziel
- Für das **Increment** ist es die Definition of Done

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

Das Product Backlog und das Sprint Backlog beschreiben auszuführende Arbeit, die Wert generieren. Das Increment wird während dem Sprint erstellt und ist ein Teil des zu entwickelnden Produktes.

## Die Scrum-Artefakte

Die drei Artefakte haben ein gemeinsames Ziel:

- Maximieren der Transparenz (Priorität, Pläne, Fortschritt)
- Fördern eines gemeinsamen Verständnisses bezüglich der Arbeit innerhalb des Scrum-Teams und mit den Stakeholdern

Neben den drei im Scrum Guide beschriebenen Artefakten gibt es noch weitere, die sich in der Praxis bewährt haben. Dies sind zum Beispiel:

- Product Vision
- Release Plan
- Taskboard
- Impediment Backlog
- Burndown Chart
- Team Backlog
- Velocity Chart
- Regeln für die Zusammenarbeit
- Themen und Themenpark



Das **Product Backlog** steigert Transparenz und Fokus bezüglich dem **Produktziel**



Das **Sprint Backlog** steigert Transparenz und Fokus bezüglich dem **Sprint Ziel**



Das **Inkrement** steigert Transparenz und Fokus bezüglich der **Definition of Done**



# Das Product Backlog

---

Die Anforderungen an das zu liefernde Produkt stehen in Scrum in einer priorisierten Liste mit dem Namen Product Backlog. Der Product Owner ist für das Product Backlog alleinig verantwortlich.

Das Product Backlog ist eine sich entwickelnde, dynamische und geordnete Liste der Dinge, die zur Produktentwicklung benötigt werden. Es ist die einzige Quelle von Arbeit, die durch das Scrum-Team erledigt wird.

Das Product Backlog umfasst die Listeneinträge und das **Produkt-Ziel**.

Product-Backlog-Einträge, die durch das Scrum-Team innerhalb eines Sprints abgeschlossen werden können (Done), gelten als bereit für die Auswahl in einem Sprint-Planning-Event. Diese Transparenzgrad erlangen sie in der Regel durch Refinement-Aktivitäten. Das Refinement des Product Backlogs ist der Vorgang, durch den Product-Backlog-Einträge in kleinere, präzisere Elemente zerlegt und weiter definiert werden. Dies ist eine kontinuierliche Aktivität, wodurch weitere Details wie Beschreibung, Reihenfolge und Größe ergänzt werden. Die Attribute variieren oft je nach Arbeitsumfeld.

Die Developer bestimmen die Größe der Product Backlog Einträge. Der Product Owner kann die Developer beeinflussen, indem er Sie dabei unterstützt, die Product-Backlog-Einträge besser zu verstehen und Kompromisse einzugehen.

Oft arbeiten mehrere Scrum-Teams gemeinsam an einem Produkt. Benutzen Sie auch hier möglichst nur ein Product Backlog und gruppieren Sie z. B. mit einem Product Backlog-Attribut, die Items nach Teams.

## **Commitment: Produkt-Ziel**

Das Produkt-Ziel beschreibt einen zukünftigen Zustand des Produkts, welches dem Scrum-Team als Planungsziel dienen kann. Das Produkt-Ziel ist das langfristige Ziel für das Scrum-Team und gehört

zum Product Backlog. Das Product Backlog entsteht, um zu definieren, „was“ das Produkt-Ziel erfüllt.

*Ein Produkt liefert Wert. Es hat klare Grenzen, bekannte Stakeholder, eindeutig definierte Benutzer oder Kunden. Ein Produkt kann eine Dienstleistung, ein physisches Produkt oder etwas Abstrakteres sein.*

### Übersicht über das Product Backlog

Im Product Backlog werden alle Features, Funktionalitäten, Verbesserungen und Fehlerbehebungen aufgelistet, welche notwendig sind ein erfolgreiches Produkt zu entwickeln.

Ich empfehle Ihnen zu Beginn nicht zu viele Anforderungen in das Product Backlog zu packen, sondern nur die wichtigsten. Generell sollte das Product Backlog so kurz und knapp wie möglich und so umfangreich und detailliert wie notwendig sein.

Nehmen Sie alle wesentlichen Anforderungen auf, die zum erfolgreichen Einsatz der Software notwendig sind. Die Frage ist natürlich immer, was ist wesentlich? Das Product Backlog sollte zu Beginn des Projektes jedoch mindestens Anforderungen für die ersten zwei Sprints enthalten.

Als Product Owner sollten Sie darauf achten, dass nicht nur zentrale Funktionen, sondern auch *nichtfunktionale Anforderungen* wie Robustheit, Skalierbarkeit, Performance usw. frühzeitig ermittelt und beschrieben werden.

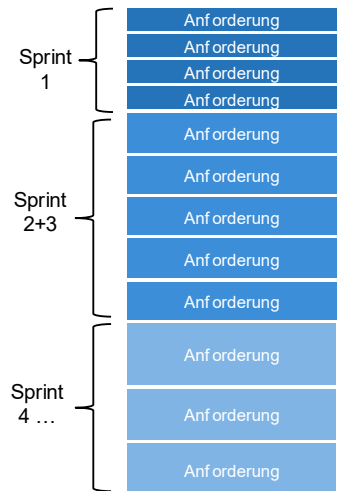


Abbildung 12: Das rangierte Product Backlog

## Stichwortverzeichnis

---

### A

Adaptive Software Development 20  
Agil oder Lean 22  
Agile Manifest 17  
Agile Methoden 15, 20  
Agile Prinzipien 18  
Agile Prinzipien und Praktiken 23  
Agilen Methoden, Unterschiede 21  
Agiles Projektmanagement 11  
Agiles Projektmanagement, Struktur 23  
Agilität und Menschen 19  
Agilität, was ist es 21  
Aktivitätenplanung 121  
Anforderung rangieren 97  
Anforderungen 14, 32, 91, 95  
Anforderungen erfassen 96  
Anforderungen, gute 98  
Anforderungen, nicht funktionale und technische 100  
Anforderungsbeschreibung 40  
Anforderungsmanagement 91, 95  
Anforderungsworkshop 96, 106  
Anpassung 29  
Anwender (User) 46  
Anwendererzählung 99  
Arbeit ausführen 121  
Arbeit priorisieren 121  
Arbeitszyklen 32  
Artefakte, weitere 34  
Auftraggeber 46  
Aufwand schätzen 80  
Aufwandschätzung 112  
Autoindustrie 8

### B

Backlog Refinement 80, 114

Burndown Chart 123, 124  
Business Value 81

### C

Commitment 75

### D

Daily Scrum 62  
Daily Scrum durchführen 63  
Daily Scrum vorbereiten 62  
Daily Scrum, verteilte Teams 65  
Definition of Done 87  
Definition of Ready 60  
Definitionsphase, trad. Projekt 95  
Developer 42

### E

Empirie 28  
empirischer Prozesssteuerung 28  
Entwicklungsgeschwindigkeit 109  
Entwicklungsgeschwindigkeit bestimmen 117  
Epics 102  
Estimation Meeting 114  
Estimation Meeting, Teilnehmer 115  
Events 53  
Explorationssprints 107  
Extreme Programming 20

### F

Feature Driven Development 20  
Fehlerkultur 52  
Fokus (Focus) 51  
Fortschritt überwachen 123  
Fortschrittsüberwachung 86

**H**

Hindernisse 65

**I**

Impediments 65

Increment **87**

Initialisierungsphase 106

Inspect &amp; Adapt 33, 62

INVEST, Kriterien 98

Iterationen 32

**J**

Jeff Sutherland 8, 17

**K**

Ken Schwaber 8, 17

klassische Projekte **14**

Kompensationssysteme 12

Kunde **46****L**

Lean Development 22

Lean Innovation 22

leichtgewichtig 13

**M**

Macht 12

Manager **47**

Marktbedürfnisse 15

Minimum Marketable Feature Set  
97, **103**Minimum Viable Product **103**

Mittelfristplanung 108, 110

MoSCoW 82

Mut (Courage) **51****N**New New Product Development  
Game 25**O**objektorientierte Programmierung  
12Offenheit (Openness) **52****P**

Planung 108

Planung in Scrum **105**

priorisieren 81

Priorisierung nach MoSCoW 82

Product Backlog 78

Product Backlog detaillieren 80

Product Backlog Management 40

Product Backlog priorisieren 81

Product Backlog, Übersicht 79

Product Increment 87

Product Owner **39**

Produktlebenszyklen 13

Produktvision **93**

Produkt-Ziel 78

Projekt steuern 127

Projektplanung 105

Projektstart 106

Proof of Concept 107

**R**Rapid Application Development  
(RAD) 13

Rational Unified Process (RUP) 13

Releasemanagement 41

Releaseplanung **108**Respekt **52**

Retrospektive 70

Retrospektive durchführen 72

Retrospektive, Ablauf 71

Retrospektive, Maßnahmen 73

Retrospektive, Maßnahmen  
priorisieren 74

Retrospektive, vorbereiten 71

Return on Investment (ROI) 48

Risiken 116

Rollen 33

Rugby-Ansatz 25

## S

Schätzen mit Punkten 115  
Schnellüberblick 31  
Scrum Artefakte 34  
Scrum Ereignisse 33  
Scrum Framework **25**  
Scrum Guide **26**  
Scrum im Schnellüberblick 31  
Scrum Master **43**  
Scrum Master, Aufgaben 43  
Scrum Workflow **128**  
Scrum, die drei Säulen 28  
Scrum, Entstehung 25  
Scrum-Artefakte 75  
Scrum-Team 33, **35**  
Scrum-Team Größe 35  
Scrum-Team, Umfeld 47  
Scrum-Werte **49, 50**  
SDLC 12  
Selbstorganisierende und  
interdisziplinäre Teams 37  
Set-Based Concurrent Engineering  
22  
Softwareentwicklungsmethoden 12  
Sprint 55  
Sprint 0 107  
Sprint abbrechen 122  
Sprint Backlog **84**  
Sprint Burndown Chart 124  
Sprint durchführen 119  
Sprint Review **66**  
Sprint Review Teilnehmer 67  
Sprint Review, Ablauf 68  
Sprintdauer, optimale 116  
Sprint-Retrospektive 70  
Sprint-Review, Feedback 68  
Sprint-Review, Vorbereitung 68  
Sprintstart, Wochentag 56  
Sprint-Umfang anpassen 122

Sprint-Ziel 66, 85, 92, 119  
Sprint-Ziel **84**  
Stakeholder 37, 69  
Systems Development Life Cycle  
12

## T

Task Board 64  
Taskboard **86**, 123, 124  
Task-Planung 121  
Tasks **102**  
Teamgröße 117  
Theme **102**  
Timeboxing **54**  
Timeline-Analyse 73  
Time-to-Market 12  
Toyota Production System 8  
Transparenz 28, 52

## U

Überprüfung 29  
Umsetzungsverantwortlich 36  
Unified Process 20  
User Stories, kleine 116  
User Story **99**

## V

Velocity 117  
Verpflichtung (Commitment) **50**  
Vision 93  
Vorprojekt 106

## W

Wasserfallmodell 13  
Wasserfall-Modell 14  
Wertesystem 49