

PROJEKTMANAGEMENT-WISSEN

# SCRUM

Agiles Projektmanagement und Scrum  
erfolgreich anwenden

Proconis Publishing



Abgestimmt mit *The Scrum Guide*™

**ROLAND WANNER**

# SCRUM



**Agiles Projektmanagement  
und Scrum 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 © 2018 Roland Wanner

ISBN: 978-1983653995

V1.4 Oktober 2018, (1. Auflage Januar 2018)

## **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.....	9
<b>1 Agiles Projektmanagement .....</b>	<b>13</b>
Agiles Projektmanagement im Überblick .....	14
Der Unterschied zwischen klassischen und agilen Projekten.....	17
Agiles Projektmanagement für alle Projekte? .....	19
Was macht Teams in der Produkte-entwicklung erfolgreich? ..	21
Das Agile Manifest .....	23
Agile Prinzipien.....	26
Agile Methoden .....	29
Die Struktur des agilen Projektmanagements .....	32
<b>2 Das Scrum Framework .....</b>	<b>35</b>
So entstand Scrum.....	35
Scrum als Management-Framework.....	38
Die drei Säulen von Scrum.....	39
Scrum im Schnellüberblick .....	42
Scrum und agile Projekte in der Unternehmens-Governance..	47
<b>3 Die Rollen in Scrum.....</b>	<b>49</b>
Der Product Owner .....	52
Das Development-Team .....	55
Der Scrum Master.....	56
Die Aufgaben des Scrum Masters.....	56
Eigenschaften und weitere Funktionen des Scrum Masters ..	60
Weitere Rollen.....	62
Das Scrum-Team und sein Umfeld .....	63
<b>4 Scrum-Werte.....</b>	<b>65</b>
Die Scrum-Werte als das Fundament .....	66
Verpflichtung (Commitment).....	67

	Mut (Courage).....	68
	Fokus (Focus).....	69
	Offenheit (Openness).....	69
	Respekt.....	70
<b>5</b>	<b>Die Scrum Ereignisse – Übersicht .....</b>	<b>71</b>
	Der Sprint .....	72
	Sprint Planning .....	74
	Das Daily Scrum .....	75
	Das Sprint Review .....	76
	Die Sprint-Retrospektive .....	78
<b>6</b>	<b>Die Scrum-Artefakte – Übersicht.....</b>	<b>79</b>
	Das Product Backlog .....	80
	Das Sprint Backlog .....	82
	Das Product Increment .....	83
	Transparenz der Artefakte .....	84
	Die "Definition of Done" (DoD).....	86
	Der Scrum Workflow .....	86
<b>7</b>	<b>Das Anforderungsmanagement .....</b>	<b>89</b>
	Von der Produktvision zum Product Backlog.....	89
	Die Produktvision.....	90
	Das Product Vision Board.....	92
	Anforderungsmanagement in Scrum .....	97
	Der Anforderungsworkshop .....	99
	Die Merkmale guter Anforderungen .....	102
	User Stories .....	103
	Nichtfunktionale Anforderungen erfassen .....	106
	Unterschied zwischen Epic, Theme, User Story und Task.....	107
	Der Business Value .....	108
<b>8</b>	<b>Das Product Backlog .....</b>	<b>111</b>
	Übersicht über das Product Backlog.....	112
	Das Product Backlog detaillieren .....	114

Den Aufwand schätzen.....	115
Das Risiko der Anforderungen bestimmen .....	116
Das Product Backlog priorisieren .....	118
Priorisierung nach MoSCoW .....	119
Priorisieren ist eine wiederkehrende Optimierungsaufgabe .	120

## **9 Planung in Scrum..... 121**

Wann startet das Projekt? .....	122
Bevor das Projekt startet.....	124
Releaseplanung.....	126
DevOps – Development and Operations .....	129
Aufwandschätzung .....	132
Aufwandschätzung auf zwei Ebenen.....	132
Das Estimation Meeting .....	134
Mit Punkten schätzen .....	138
Planning Poker .....	139
Magic Estimation.....	140
User Stories sollten „klein“ sein .....	142
Die optimale Sprintdauer.....	143
Die Entwicklungsgeschwindigkeit bestimmen.....	144

## **10 Das Sprint Planning..... 147**

Der Sprint-Ablauf.....	147
Die „Definition of Ready“ .....	151
Die “Definition of Done” (DoD).....	153
Vorbereitung auf das Sprint Planning.....	155
Sprint Planning 1 .....	156
Sprint Planning 2 .....	160

## **11 Die Sprint-Durchführung..... 163**

Artefakte und Ereignisse im Sprintablauf .....	163
Ausführen der Arbeit .....	165
Das Sprint Backlog .....	166
Das Scrum Taskboard.....	168

<b>12</b>	<b>Das Daily Scrum .....</b>	<b>171</b>
	Das Daily Scrum vorbereiten und durchführen.....	171
	Den Sprint-Umfang anpassen.....	175
	Einen Sprint abbrechen.....	177
	Den Fortschritt überwachen.....	178
	Das Taskboard.....	179
	Das Sprint Burndown Chart.....	179
	Das Burnup Chart.....	181
	Das Story Burndown Chart.....	182
	Der Themenpark.....	183
	Das Velocity Chart.....	184
<b>13</b>	<b>Das Sprint Review .....</b>	<b>187</b>
	Das Ziel des Sprint Reviews.....	187
	Vorbereitung und Ablauf des Sprint- Reviews.....	188
	Das Resultat des Sprint Reviews.....	191
<b>14</b>	<b>Die Sprint Retrospektive .....</b>	<b>193</b>
	Kontinuierliche Verbesserung und stetiges Lernen.....	193
	Vorbereiten der Retrospektive.....	195
	Die Retrospektive durchführen.....	197
	Sicherheit schaffen.....	197
	Ereignisse visualisieren.....	198
	Erkenntnisse sammeln.....	199
	Maßnahmen definieren.....	200
	Maßnahmen priorisieren.....	201
<b>15</b>	<b>Zusammenarbeit im Scrum-Team .....</b>	<b>205</b>
	Die neue Team-Philosophie.....	206
	Große Herausforderungen.....	208
	Fest zugeordnetes Team.....	208
	Limitierte Teamgröße.....	209
	Selbstorganisierende Teams.....	209
	Funktionsübergreifend arbeiten.....	213
	Kontinuierliche Verbesserung und Lernen.....	214

Ownership übernehmen .....	215
Die Zusammenarbeit im Team .....	216
Zusammenarbeit mit dem Product Owner .....	216
Zusammenarbeit mit dem Scrum Master .....	217
Zusammenarbeit mit dem Auftraggeber .....	220
<b>16 Scrum bei großen Projekten .....</b>	<b>223</b>
Große Projekte und verteilte Teams.....	223
Der agile Blueprint für große Projekte .....	226
Der Scrum@Scale™ Guide.....	227
Scrum@Scale Übersicht.....	227
Der Scrum Master Cycle.....	229
Der Product Owner Cycle.....	235
Die Organisation großer und verteilter Projekte.....	240
Die Praktiken für große Projekte anpassen.....	241
Das Product Backlog bei großen Projekten.....	241
Releaseplanung bei großen Projekten .....	242
Den Aufwand bestimmen.....	242
Projektweites Sprint Review und Retrospektive .....	243
<b>17 Glossar .....</b>	<b>245</b>
<b>18 Anhang.....</b>	<b>257</b>
Internet-Links und empfohlene Literatur .....	257
Über den Autor .....	259
Literaturverzeichnis .....	260
Stichwortverzeichnis.....	262





Dieses Buch basiert auf dem **Scrum Guide™** vom November 2017 von Ken Schwaber und Jeff Sutherland sowie auf dem **Scrum@Scale™ Guide** von Jeff Sutherland von Scrum Inc. vom August 2018

# 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. Diese 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 im 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“. Dies auch mit dem Ziel, die Arbeit sinnvoller, interessanter und lohnenswerter zu gestalten. Wenn Sie schon etwas älter sind, dann kommen Ihnen diese Themen vielleicht bekannt vor, denn sie wurden teilweise schon in den 1960er Jahren eingesetzt.

## **Erfolgreiche Managementmethoden wiederentdeckt**

Die Prinzipien, Methoden und Werte, auf denen das agilen Projektmanagement aufbaut, gibt es schon seit Jahrzehnten. Ich habe mich

schon vor 30 Jahren damit beschäftigt. Sie stammen praktisch alle aus den 1950er und 1960er Jahren vom Toyota Production System und anderen japanischen Unternehmen. Daraus entstanden dann in den Folgejahren Management-Methoden, wie z.B. Lean Production, Lean Management, Business Process Reengineering, Six-Sigma oder das Wissensmanagement. Auch selbstorganisierende Teams sind ein alter Hut. Auch diese wurden schon vor vielen Jahrzehnten in japanischen und in einigen amerikanischen Unternehmen erfolgreich 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 sequentiell in Phasen ausgeführt, die aufeinander aufbauen. Als die Informationstechnologie und Softwareentwicklung in den 1960er Jahren entstand wurde die sequentielle Vorgehensweise auch in diese Projekte übernommen. In den Folgejahren wurden verschiedene mehr oder weniger erfolgreiche iterative Softwareentwicklungs-Methoden entwickelt.

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 agilen 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 ist. 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 **ab Seite 42** einen schnellen **Überblick über das Scrum Framework**, damit erhalten Sie innerhalb von fünf Minuten ein Grundwissen und verstehen so später die Zusammenhänge viel besser.

Die folgenden Kapitel geben Ihnen einen **Überblick** über die Werte von Scrum, Scrum Ereignisse (Events) und Artefakt. Die weiteren Kapitel vertiefen dann das Gelernte und gehen ins Detail der Scrum Artefakte und Ereignisse.

Sie werden während dem Lesen sicher feststellen, dass viele Themen in diesem Buch mehrfach in verschiedener Granularität vorkommen. Das ist kein Fehler oder nur um das Buch mit Seiten zu füllen, sondern dient dazu, damit Sie sich immer weiter in Scrum vertiefen und immer mehr Details der verschiedenen Themen kennenlernen.

### **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. Sie wird periodisch aktualisiert. Dieses Buch basiert auf der neusten Version des Scrum Guides vom November 2017 und auf dem Scrum@Scale Guide vom Februar 2018.

Der Scrum Guide beschreibt in sehr knapper Form die Mindestanforderung an Scrum. Er kann hier kostenlos heruntergeladen werden: <https://www.scrum.org/resources/scrum-guide>

### Das Agile Manifest

Das Agile Manifest für Softwareentwicklung ist der kleinste gemeinsame Nenner aller agilen Vorgehensmodelle – und damit auch für Scrum. Es beschreibt die „Zwölf Prinzipien Agiler Softwareentwicklung“, die Sie hier nachlesen können:

<http://agilemanifesto.org/>

### Terminologie in diesem Buch

Ich verwende in diesem Buch bewusst die bekannten, englischsprachigen Scrum-Begriffe für die *Rollen*, *Artefakte* und *Ereignisse*. 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.

# 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.

Seit kurzem wird versucht agile Methoden auch in anderen Geschäftsprozessen einzusetzen als bei Projekten. Dies ist eine spannende Herausforderung, welche die gesamte Unternehmenskultur, Führungssysteme und die Zusammenarbeit stark beeinflussen wird. Ich bin mir aber nicht so sicher, ob viele Großunternehmen diesen Schritt in den nächsten Jahren schaffen werden. Kleinunternehmen sind diesbezüglich viel anpassungsfähiger.

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 und werden viele Unternehmen noch Mühe haben.

## Agiles Projektmanagement

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.

Das Prinzip von selbstorganisierenden und funktionsübergreifenden Teams, Chefs als Coaches ohne Führungsfunktion und der Abbau von Managementebenen, war schon vor Jahrzehnten kurz ein „interessantes Thema“. Ich hoffe in den nächsten Jahren haben wir damit mehr Erfolg.

In der Softwareentwicklung mit Scrum werden agile Methoden schon seit einigen 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.

Software und Softwareprojekte gibt es jedoch erst seit ein paar Jahrzehnten. In den 1950er Jahren war Software noch Teil der Hardware und wurde als Programmcode bezeichnet. Ich kann mich auch noch gut an die Lochkarten erinnern, mit denen in den 1970er Jahren Werkzeugmaschinen gesteuert wurden. Die Lochkarten waren die

Programme, die notwendig waren, damit Werkzeugmaschinen z.B. Teile fräsen konnten.

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 sequentiellen Wasserfallmodell abgewickelt (Abbildung Seite 17). 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 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:

- 1986 – Barry Boehm entwickelte erste Ansätze eines iterativen Software-Entwicklungsprozesses mit dem risikoorientierten Spiralmodell.



## Agiles Projektmanagement

- 1991 – Rapid Application Development (RAD) wurde vorgestellt
- 1995 – Jeff Sutherland und Ken Schwaber präsentierten zusammen ein Dokument, welches die Scrum Methode erstmals beschrieb
- 1998 – Rational Unified Process (RUP) wurde vorgestellt
- 1999 – Extreme Programming (XP) wurde vorgestellt, das auf großes Interesse bei den Softwareentwicklern stieß.

Wie Sie sehen, sind erste Ansätze zu agiler Softwareentwicklung bereits Anfang der 1990er Jahre zu finden. Popularität erreichte die agile Softwareentwicklung erstmals 1999, als Kent Beck das erste Buch zu Extreme Programming (XP) veröffentlichte.

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 23) 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 das 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 und sich so schnell an Veränderungen anzupassen, 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.

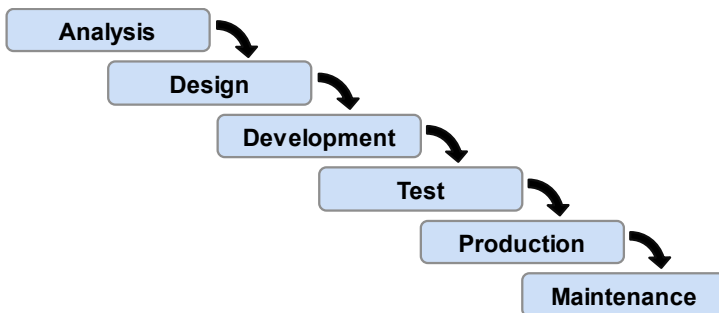


Abbildung 1: Das Wasserfall-Modell

Sie können sich vermutlich gut vorstellen wie man ein Haus baut. Wenn bei diesem Projekt die Bauarbeiter bereits die Wände mauern

## Agiles Projektmanagement

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 u.a.

- auf enge Zusammenarbeit zwischen 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 (Iterationen mit definierter Länge; typisch sind 14 bis maximal 30 Tage).

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

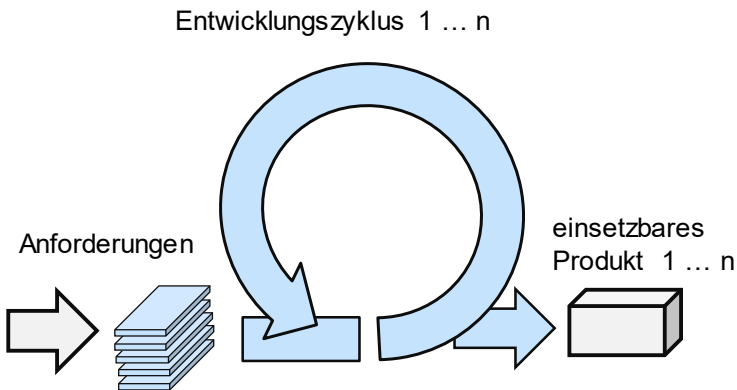


Abbildung 2: Agiles Vorgehen

### Agiles Projektmanagement für alle Projekte?

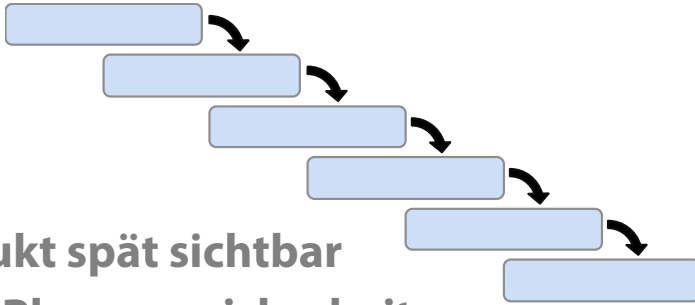
Agiles Projektmanagement setzt sich immer mehr durch, auch außerhalb der traditionellen Softwareentwicklung. Viele sagen dem Wasserfallmodell schon den Tod voraus. Soweit wird es aber nicht kommen. Versuchen Sie ein Haus, Flugzeugträger, Werkzeugmaschine oder eine Fabrikationsanlage mit agilem Projektmanagement zu entwickeln und zu fertigen. Unmöglich! Dort müssen praktisch alle Anforderungen zu Projektbeginn bekannt sein, und der Spielraum diese während dem Projektablauf zu ändern oder neue hinzuzufügen ist sehr gering.

Bei der Entwicklung von „physischen Produkten“ wird das Wasserfall-Modell der bevorzugte Projektmanagement-Prozess bleiben, bei „immateriellen Produkten“ wie z.B. Software wird sich agiles Projektmanagement aber immer mehr durchsetzen.

Das heißt jedoch nicht, dass agile Methoden nicht für bestimmte Lieferobjekte von „Wasserfallprojekten“ eingesetzt werden können. Überall, wo zum Beispiel Software entwickelt werden muss, können wahrscheinlich agile Entwicklungsmethoden eingesetzt werden, zum Beispiel für die Software einer Werkzeugmaschine oder eines Car Entertainment Systems.

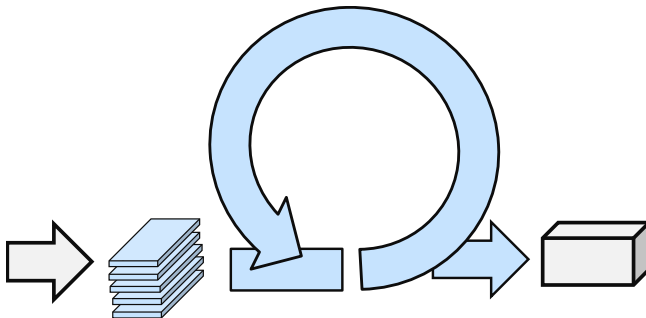
Bestimmte Prinzipien, Praktiken und Werte des agilen Projektmanagements werden sicher auch das traditionelle Projektmanagement in Zukunft stark beeinflussen. Ich denke nur schon an die Zusammenarbeit im Team und an die agilen Werte und Prinzipien. Aber überall wo physische Produkte entwickelt werden ist die Flexibilität im Entwicklungsprozess eindeutig geringer als in der Softwareentwicklung.

# Wasserfall-Modell



- **starr**
- **Produkt spät sichtbar**
- **hohe Planungssicherheit**
- **konstante Anforderungen notwendig**

# Agiles Vorgehen



- **wenig Struktur, iterativ**
- **früh einsetzbare Resultate**
- **hohe Flexibilität, marktnah**
- **ändernde Anforderungen willkommen**

## Was macht Teams in der Produktentwicklung erfolgreich?

---

Ein Kernpunkt im agilen Projektmanagement ist die Zusammenarbeit in Teams und das Anwenden von Wissen. Um mehr darüber zu erfahren machen wir jetzt einen kurzen Abstecher in die Vergangenheit.

Hiroataka Takeuchi und Ikujiro Nonaka beschrieben 1986 im Harvard Business Review Artikel "The New New Product Development Game" die Eigenschaften erfolgreicher Produktentwicklungsteams. Der Inhalt des Artikels ist stark beeinflusst vom „Toyota Production System“ und japanischen Produktentwicklungsmethoden z.B. bei Canon, Honda oder NEC aus den 1970er Jahren. In diesem Artikel werden wesentliche Eigenschaften von erfolgreichen Product-Development-Teams beschrieben:

1. **Built-in instability:** Das Management stößt den Entwicklungsprozess an, setzt herausfordernde Ziele und Anforderungen und gibt dem Team größtmögliche Freiheit in der Entwicklung.
2. **Self-organizing project teams:** Das Team arbeitet wie ein Start-up Unternehmen, es ist initiativ, geht Risiken ein und hat eine unabhängige Agenda: Die Teams charakterisieren sich durch folgende drei Eigenschaften:
  - *Autonomy:* Die Teams organisieren sich selber.
  - *Self-transcendence:* Stetiges lernen. Das Team strebt nach vorne und will immer besser werden.
  - *Cross-fertilization:* Die Teams arbeiten funktionsübergreifend. Diese Diversität fördert neue Ideen und Konzepte, was zu erfolgreicherer Produkten führt.
3. **Overlapping development phases:** Mit stark überlappenden Phasen wird der Entwicklungsprozess schneller und flexibler.

4. **Multilearning:** Durch Lernen auf verschiedenen Unternehmensebenen und in Gruppen, durch stetigen Erfahrungsaustausch und durch den Kontakt mit der Umwelt kann man schneller auf verändernde Marktbedingungen reagieren.
5. **Subtle control:** Das Management setzt genügend Checkpunkte, obwohl sich das Projektteam selbst organisiert. Damit sollen Instabilität, Unklarheiten und Spannungen vermieden werden.
6. **Organizational transfer of learning:** Gelerntes Wissen soll in die Organisation einfließen, um sich stetig zu verbessern.

Außerhalb Japans haben nur wenige Industrien Elemente dieses Konzepts in ihre Geschäftstätigkeit einließen lassen, und wenn, dann oft eher halbherzig. Abgeleitet aus diesem Wissen entstand in den 1990er Jahren das Lean Management und das Knowledge Management.

Die geringe Erfolgsquote bei Softwareprojekten in den 1990er Jahren lag nicht in der Ausbildung der Mitarbeiter oder an der fehlenden Reife, der noch jungen Informatik und deren Werkzeugen, sondern in der Durchführung und der Zusammenarbeit in Projekten. Ken Schwaber und Jeff Shuterland haben dies erkannt und dann aus dem Wissen von "The New New Product Development Game" und eigenen Erfahrungen ein erfolgreiches Softwareentwicklungs-Framework entwickelt – SCRUM.

### Was hat Scrum mit Rugby zu tun?

Sie haben sich vielleicht gewundert, was Scrum mit Rugby zu tun hat. Das Titelbild dieses Buches zeigt ein „Scrum“ beim Rugby. Dies ist das Gedränge der Spieler beim Einwurf des Balles nach kleineren Regelverstößen.

Jeff Sutherland und Ken Schwaber haben sich vom Business Review Artikel "The New New Product Development Game" inspirieren lassen, indem „Rugby“ und der Zusammenhang zu erfolgreichen, selbstorganisierenden Teams mehrmals erwähnt wurde. Darum benutzten Jeff und Ken für ihr neues Softwareentwicklungs-Framework einen Begriff aus dem Rugby – und dieser ist SCRUM.

# Das Agile Manifest

---

Im Frühjahr 2001 verabschiedeten 17 projekterfahrene Software-Entwickler in Utah (USA) das sogenannte „Manifesto for Agile Software Development“, heute vor allem unter der Kurzbezeichnung „Agile Manifesto“ bekannt. Diese Erstunterzeichner, darunter auch die beiden Scrum-Gründer Ken Schwaber und Jeff Sutherland, formulierten mit dem Agile Manifesto erstmals die zentralen Werte agiler Softwareentwicklung – ein Meilenstein und zugleich das Fundament des agilen Projektmanagements.

Die Werte des Agilen Manifests ([www.agilemanifesto.org](http://www.agilemanifesto.org)) sind paarweise beschrieben, wobei die Werte auf der linken Seite jeweils höher eingeschätzt werden, als die Werte auf der rechten Seite. Dies heißt aber nicht, dass diese bedeutungslos sind.

### Das Agile Manifest lautet folgendermaßen:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte schätzen gelernt:

- **Individuen und Interaktionen** stehen über Prozessen und Werkzeugen
- **Funktionierende Software** steht über einer umfassenden Dokumentation
- **Zusammenarbeit mit dem Kunden** steht über der Vertragsverhandlung
- **Reagieren auf Veränderung** steht über dem Befolgen eines Plans

„Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Ich würde die agilen Werte des Agilen Manifests eher als „Grundsätze der agilen Softwareentwicklung bezeichnen“, damit diese nicht mit den Scrum-Werten verwechselt werden.



# Anhang

## Internet-Links und empfohlene Literatur

---

Auf der Internetseite <http://www.rolandwanner.ch> finden Sie eine Liste mit Links und Artikel zum agilen Projektmanagement, Projektcontrolling, Earned Value Management und Risikomanagement in Projekten.

Auf meinem Blog <http://www.rolandwanner.ch/blog> finden Sie interessante Artikel zum agilen Projektmanagement, Projektcontrolling, Earned Value Management und Risikomanagement in Projekten.

### **Folgende ergänzenden Scrum-Bücher empfehle ich:**

Scrum in der Praxis: Erfahrungen, Problemfelder und Erfolgsfaktoren, Sven Röpstroff, Robert Wiechmann, dpunkt.verlag, 2015

Scrum: Produkte zuverlässig und schnell entwickeln, Boris Gloger, Hanser Verlag, 2016

Agiles Produktmanagement mit Scrum: Erfolgreich als Product Owner arbeiten, Roman Pichler, dpunkt.verlag, 2013

### **Folgende Blogs empfehle ich zu verfolgen:**

Mike Cohn's Blog, <https://www.mountaingoatsoftware.com/blog>

Scrum Alliance, Resources : <https://www.scrumalliance.org/agile-resources>

Roman Pichler's Blog: <http://www.romanpichler.com/blog/>

Scrum.org Blog: <https://www.scrum.org/resources/blog>

scrum inc. Blog: <https://www.scruminc.com/scrum-blog/>

Innolution Blog (Kenneth S. Rubin): <https://innolution.com/blog>

LeadingAgile Blog: <https://www.leadingagile.com/blog/>

DZone Agile Zone Articles: <https://dzone.com/agile-methodology-training-tools-news>

AgileConnection Blog : <https://www.agileconnection.com/>

## Über den Autor

---

Roland Wanner ist schon seit über 30 Jahren im Projektgeschäft tätig und hat schon viele Projekte miterlebt – erfolgreiche und gescheiterte. Nach seiner Ausbildung als Maschineningenieur und Wirtschaftsingenieur war er zuerst 5 Jahre als Projektleiter und dann mehrere Jahre als Projektcontroller und Projektportfolio-Manager im Maschinen- und Anlagenbau tätig. Seit mehr als 10 Jahren arbeitet er als Projektmanagement-Spezialist, Projektportfolio-Manager und Project Office Manager im Banken- und Versicherungsbereich. Dort unterstützte er verschiedene Softwareprojekte, die Scrum und agiles Projektmanagement angewendet haben.

Auf seinem Blog <http://www.rolandwanner.ch/blog> finden Sie interessante Artikel über agiles Projektmanagement, Projektcontrolling, Earned Value Management und Risikomanagement in Projekten.

### **Ihre Meinung ist uns wichtig!**

Herzlichen Dank, dass Sie dieses Buch gekauft haben. Wir haben unser Bestes gegeben, beim Inhalt wie auch bei der Aufmachung. Es wurde viel Aufwand geleistet, um dieses Buch so vollständig und korrekt wie möglich zu machen. Es ist jedoch nicht ganz auszuschließen, dass uns an der einen oder anderen Stelle des Buches ein Missgeschick unterlaufen ist, ob inhaltlich oder in der Rechtschreibung. Vielleicht vermissen Sie auch bestimmte Informationen oder sind der Meinung, gewisse Themen sollten vertieft werden, oder sind bei gewissen Themen anderer Meinung. Wir sind auf Ihre Meinung angewiesen!

Für Ihre Ideen, Gedanken und Korrekturvorschläge bedanken wir uns ganz herzlich. Senden Sie diese bitte an: [info@rolandwanner.ch](mailto:info@rolandwanner.ch)

## Literaturverzeichnis

- Amabile, T. (2011). *The Progress Principle: Using Small Wins to Ignite Joy, Engagement, and Creativity at Work*. Harvard Business Review Press.
- Cohn, M. (2012). *Release Planning: Retiring the Term but not the Technique*. Von <https://www.mountangoatsoftware.com/blog/release-planning-retiring-term-not-technique> abgerufen
- Cohn, M. (2014). *Agile Estimating and Planning*. Prentice Hall.
- Cohn, M. (2016). *Don't Estimate the Sprint Backlog Using Task Points*. Von <https://www.mountangoatsoftware.com/blog/dont-estimate-the-sprint-backlog-using-task-points> abgerufen
- Cohn, M. (2017). *Planning Poker Cards: Effective Agile Planning and Estimation*. Von <https://www.mountangoatsoftware.com/tools/planning-poker> abgerufen
- Cohn, M. (2017). *When Should We Estimate the Product Backlog*. Von <https://www.mountangoatsoftware.com/blog/when-should-we-estimate-the-product-backlog> abgerufen
- Dräther, R. (2013). *Scrum kurz & gut*. O'Reilly.
- Gloger, B. (2016). *Scrum - Produkte zuverlässig und schnell entwickeln*. Hanser.
- Gloger, B. (2017). *Was ist Scrum*. Von <https://borisgloger.com/scrum/> abgerufen
- Greaves, K. (2012). *Release Planning with Scrum*. Von <https://www.growingagile.co.za/2012/10/release-planning-with-scrum/> abgerufen
- Joas, H. (1999). *Die Entstehung der Werte*. Suhrkamp.
- Kerth, N. (2017). *The Retrospective Prime Directive*. Von <http://retrospectives.com/pages/retroPrimeDirective.html> abgerufen
- McChrystal, G. S. (2015). *Team of Teams: New Rules of Engagement for a Complex World*. Penguin.

- Pichler, R. (2011). *The Product Vision Board*. Von <https://www.romanpichler.com/blog/the-product-vision-board/> abgerufen
- Pichler, R. (2013). *Scrum - Agiles Projektmanagement erfolgreich einsetzen*. dpunkt.Verlag.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press.
- Scrum.de. (2015). *Scrum.de*. Von <https://www.scrum.de/wir-brauchen-devops/> abgerufen
- Wintersteiger, A. (2012). *Scrum Schnelleinstieg*. entwickler.press.

# Stichwortverzeichnis

## 3

3C Kriterien 105

## A

Adaptive Software Development 29  
Agil oder Lean 31  
Agile Frameworks 226  
Agile Manifest 12, **23**  
Agile Manifests - Bedeutung 25  
Agile Methoden 18, **29**  
Agile Prinzipien **26**  
Agile Werte **65**  
Agilen Methoden, Unterschiede 30  
Agiles Projektmanagement 13  
Agiles Projektmanagement, Struktur **32**  
Agilität im Projektmanagement 25  
Agilität und Menschen 27  
Agilität und Wissen 27  
Agilität, was ist 30  
Aktivitäten definieren 161  
Anforderung rangieren 100  
Anforderungen 17, 43, 89, 97  
Anforderungen erfassen 99  
Anforderungen verstehen 100  
Anforderungen, detaillierte 157  
Anforderungen, gute 102  
Anforderungen, High-Level 94  
Anforderungen, nicht funktionale und technische 106  
Anforderungen, nicht funktionale, funktionale 94  
Anforderungsbeschreibung 53  
Anforderungsmanagement **89, 97**  
Anforderungsworkshop 90, 99, 101, 111  
Anpassung 40  
Anwender (User) **62**

Anwendererzählung 103  
Arbeit ausführen 165  
Arbeit priorisieren 165  
Arbeitszyklen 43  
Architektur und Design 160  
Artefakt Transparency 80  
Artefakte, weitere 45  
Artifact Transparency 46  
Auftraggeber **62**  
Aufwand schätzen 115  
Aufwandschätzung **132**  
Autoindustrie 10  
Automatisierung 130

## B

Backlog Refinement 81, 134, 235  
Barry Boehm 15  
Burndown Chart 178, **179**  
Burnup Charts 181  
Business Case 47, 122  
Business Process Engineering 10  
Business Value **108**

## C

Chief Product Owner 231, 236  
Coach 60  
CPO *Siehe* Chief Product Owner

## D

Daily Scrum 75, **171**  
Daily Scrum durchführen 172  
Daily Scrum vorbereiten 171  
Daily Scrum, verteilte Teams 174  
DEEP 113  
Definition of Done 86, **153**  
Definition of Ready **151**  
Definitionsphase, trad. Projekt 97  
Development and Operations 129

Development-Team **55**  
 DevOps 129, 130  
 Die Rollen in Scrum 44, **49**  
 Distributed Teams 224  
 Done 153

## E

Empirie 39  
 empirischer Prozesssteuerung 39  
 Entwicklungsgeschwindigkeit 127,  
 184  
 Entwicklungsgeschwindigkeit  
 bestimmen 144  
 Epics **107**  
 Ereignisse **71**  
 Estimation Meeting 112, 114, **134**  
 Events **71**  
 Executive Action Team (EAT) 233  
**Executive MetaScrum** 238  
 Explorationssprints 124  
 externe Projekte 122  
 Extreme Programming 29  
 Extreme Programming (XP) 16

## F

Feature Driven Development 29  
 Feedback 190  
 Fehlerkultur 70  
 Fokus (Focus) **69**  
 Fortschritt überwachen **178**  
 Fortschrittsüberwachung 167  
 Führungssysteme 13  
 Funktionsübergreifend arbeiten 213

## G

große Projekte 223  
 Großunternehmen 13  
 Grundursachen, finden 200

## H

Hindernisse 174

## I

Impediment Backlog 202  
 Impediments 174  
 Initialisierungsphase 124  
 Inspect & Adapt 44, 72, 75, 194  
 Iterationen 43

## J

japanischen  
 Produkteentwicklungsmethoden  
 21  
 Jeff Sutherland 10, 23, 226

## K

Karteikarten 103  
 Ken Schwaber 10, 23, 226  
 klassische Projekte 17  
 Kleinunternehmen 13  
 Knowledge Management 22  
 Kompensationssysteme 14, 206,  
 219  
 Konflikte im Team 218  
 kontinuierliche Verbesserung 193  
 kontinuierlichen  
 Verbesserungsprozess 194  
 Kunde **62**  
 Kundenanforderungen verstehen  
 100  
 Kundenbedürfnisse 94

## L

Large Scale Scrum 226  
 Lean Development 31  
 Lean Innovation 31  
 Lean Management 10, 22  
 Lean Production 10  
 leichtgewichtig 16  
 Leistungsbewertung 219  
 LeSS 226

## M

Macht 14, 206

# Anhang

Magic Estimation 140  
Manager **63**  
Marktbedürfnisse 18  
Märkte 94  
Marktsegment 94  
Mentor 60  
Mentor und Coach 217  
MetaScrum 235  
Minimum Marketable Feature Set  
100, **108**  
Minimum Viable Product **109**  
Mittelfristplanung 126, 128  
Mut (Courage) **68**

## N

Net Present Value 108  
New New Product Development  
Game 35

## O

objektorientierte Programmierung  
15  
Offenheit (Openness) **69**  
Ownership übernehmen 215

## P

Parking Lot Diagram 183  
Pflichten- und Lastenheft 122  
Pflichtlieferobjekte 47  
Planning Poker 139  
Planung 126  
Planung in Scrum **121**  
Pre-Sprint Story Review 155  
Priorisierung nach MoSCoW 119  
Product Backlog 80, **111**  
Product Backlog bei großen  
Projekten 241  
Product Backlog detaillieren 81  
Product Backlog Management 53  
Product Backlog priorisieren 118  
Product Backlog, detaillieren 114  
Product Backlog, Übersicht 112  
Product Increment 83  
Product Owner **52**

**Product Owner Cycle** 235  
Product Vision Board 92  
Product-Development-Teams 21  
Produktattribute 94  
Produkteentwicklungsteams 21  
Produktlebenszyklen 15  
Produktvision **90**  
Produktvision teilen,  
kommunizieren 95  
Projektbudget 122  
Projekt-Governance 47  
Projektorganisation, große Projekte  
240  
Projektplanung 121  
Projektstart 122  
Projekt-Startaktivitäten 122  
Proof of Concept 125

## R

Rapid Application Development  
(RAD) 16  
Rational Unified Process (RUP) 16  
Release-Management 128  
Releaseplanung **126**  
Releaseplanung bei großen  
Projekten 242  
Remote Teams 224  
Respekt **70**  
Retrospektive durchführen 197  
Retrospektive, Ablauf 196  
Retrospektive, Maßnahmen 200  
Return of Investment 108  
Return on Investment (ROI) 63  
Risiken 143  
Risiko der Anforderungen 116  
Rugby 22  
Rugby-Ansatz 35

## S

SAFe® 209, 226  
ScALed 209, 226  
Scaled Agile Framework 226  
Scaled Agile Lean Development  
226  
Scaled Daily Scrum 230



- Schätzen mit Punkten 138
  - Scope Änderungen 175
  - Scrum als Management Framework 38
  - Scrum Artefakte 45
  - Scrum Artefakte – Übersicht 79
  - Scrum bei großen Projekten 223
  - Scrum But 46
  - Scrum Ereignisse 44, **71**
  - Scrum Framework **35**
  - Scrum Guide 12, **36**
  - Scrum im Schnellüberblick 42
  - Scrum Master **56**
  - Scrum Master Cycle 229
  - Scrum Master Cycle, Ergebnisse 233
  - Scrum Master, Aufgaben 56
  - Scrum Master, Eigenschaften 60
  - Scrum Master, Entscheidungen 60
  - Scrum of Scrums 229
  - Scrum of Scrums Master (SoSM) 231
  - Scrum Taskboard 168
  - Scrum Team, Umfeld 63
  - Scrum Workflow **86**
  - Scrum, die drei Säulen 39
  - Scrum, Entstehung 35
  - Scrum@Scale 227
  - Scrum@Scale Guide 227
  - Scrum-Werte **66**
  - SDLC 15
  - selbstorganisierende Teams 14, 209
  - Selbstorganisierende Teams **209**
  - Selbstorganisierende und interdisziplinäre Teams 51
  - Servant Leader 56
  - Six-Sigma 10
  - Skalierung 232
  - Softwareentwicklungsmethoden 15
  - SoS skalieren 232
  - Spiralmodell 15
  - Sprint 72
  - Sprint 0 124
  - Sprint abbrechen 177
  - Sprint Backlog 82, **166**
  - Sprint Burndown Chart 179
  - Sprint Planning 74, **147**
  - Sprint Planning 1 **156**
  - Sprint Planning 2 **160**
  - Sprint Planning, Vorbereitung 155
  - Sprint Retrospektive 78, **193**
  - Sprint Review 76, **187**
  - Sprint Review Teilnehmer 188
  - Sprint Review, Ablauf 189
  - Sprint Review, Maßnahmen daraus 191
  - Sprint Review, Resultat 191
  - Sprint-Ablauf 147
  - Sprintdauer, optimale 143
  - Sprint-Durchführung **163**
  - Sprint-Review, Vorbereitung 189
  - Sprint-Reviews, Projektweite 243
  - Sprintstart, Wochentag 150
  - Sprint-Umfang anpassen 175
  - Sprintziel 156
  - Stakeholder 49, 89, **187**, 190
  - Startaktivitäten 122
  - stetige Lernen 193
  - Story Burndown Chart 182
  - Systems Development Life Cycle 15
- ## T
- Taskboard **168**, 170, 178, 179
  - Task-Planung 165
  - Tasks **107**
  - Teambildungsprozess, Tuckman 220
  - Teamgröße 144
  - Teamkapazität überprüfen 162
  - Teamleistung und Belohnung 219
  - Team-Philosophie 206
  - Teams, mehrere 60
  - The New New Product Development Game 21, 22, 207
  - Theme **107**
  - Themenpark 183
  - Timeboxing **72**
  - Timeline 198
  - Time-to-Market 15
  - Toyota Production System 10, 21

## Anhang

traditionelle Projektmanagement 19  
traditionellen Softwareentwicklung  
19  
Transparenz 40, 69, 84  
Transparenz der Artefakte 84

### U

Überprüfung 40  
Umfangänderungen 175  
Unified Process 29  
Unternehmens-Governance 47  
Unternehmenskultur 13  
User Stories 90  
User Stories, kleine 142  
User Story **103**

### V

Velocity 144  
Velocity Chart 184  
Verpflichtung (Commitment) **67**  
verteilte Teams 223  
Vorprojekt 124

### W

Wasserfallmodell 15, 19  
Wasserfall-Modell 17  
Wertesystem 25, 65  
Wissensmanagement 10

### Z

Zusammenarbeit im Scrum-Team  
**205**  
Zusammenarbeit in Teams 21