

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

Internet: 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

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 © 2019 Roland Wanner

ISBN: 978-1983653995

V1.5 April 2019, (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
1 Agiles Projektmanagement	15
Agiles Projektmanagement im Überblick	16
Der Unterschied zwischen klassischen und agilen Projekten.....	19
Agiles Projektmanagement für alle Projekte?	21
Was macht Teams in der Produkteentwicklung erfolgreich?...23	
Das Agile Manifest	25
Agile Prinzipien.....	28
Agile Methoden	31
Die Struktur des agilen Projektmanagements	34
2 Das Scrum Framework	37
So entstand Scrum.....	37
Scrum als Management-Framework.....	40
Die drei Säulen von Scrum.....	41
Scrum im Schnellüberblick	44
Scrum und agile Projekte in der Unternehmens-Governance..	49
3 Die Rollen in Scrum	51
Der Product Owner	54
Das Development-Team	57
Der Scrum Master.....	58
Die Aufgaben des Scrum Masters.....	58
Eigenschaften und weitere Funktionen des Scrum Masters	62
Weitere Rollen.....	64
Das Scrum-Team und sein Umfeld	65
4 Scrum-Werte	67
Die Scrum-Werte als das Fundament	68
Verpflichtung (Commitment).....	69

	Mut (Courage)	70
	Fokus (Focus).....	71
	Offenheit (Openness).....	71
	Respekt	72
5	Die Scrum Ereignisse – Übersicht	73
	Der Sprint	74
	Sprint Planning	76
	Das Daily Scrum	77
	Das Sprint Review	78
	Die Sprint-Retrospektive	80
6	Die Scrum-Artefakte – Übersicht	81
	Das Product Backlog	82
	Das Sprint Backlog	84
	Das Product Increment	85
	Transparenz der Artefakte	86
	Die "Definition of Done" (DoD)	88
	Der Scrum Workflow	88
7	Das Anforderungsmanagement	91
	Von der Produktvision zum Product Backlog.....	91
	Die Produktvision.....	92
	Das Product Vision Board.....	94
	Anforderungsmanagement in Scrum	99
	Der Anforderungsworkshop	101
	Die Merkmale guter Anforderungen	104
	User Stories	105
	Nichtfunktionale Anforderungen erfassen	108
	Unterschied zwischen Epic, Theme, User Story und Task.....	109
	Der Business Value	110
8	Das Product Backlog	113
	Übersicht über das Product Backlog.....	114
	Das Product Backlog detaillieren	116

Den Aufwand schätzen.....	117
Das Risiko der Anforderungen bestimmen	118
Das Product Backlog priorisieren	120
Priorisierung nach MoSCoW	121
Priorisieren ist eine wiederkehrende Optimierungsaufgabe .	122

9 Planung in Scrum 123

Wann startet das Projekt?	124
Bevor das Projekt startet.....	126
Releaseplanung.....	128
DevOps – Development and Operations	131
Aufwandschätzung	135
Aufwandschätzung auf zwei Ebenen.....	135
Das Estimation Meeting	137
Mit Punkten schätzen	141
Planning Poker	142
Magic Estimation.....	143
User Stories sollten „klein“ sein.....	145
Die optimale Sprintdauer.....	146
Die Entwicklungsgeschwindigkeit bestimmen.....	147

10 Das Sprint Planning 149

Der Sprint-Ablauf.....	149
Die „Definition of Ready“	153
Die “Definition of Done” (DoD).....	155
Vorbereitung auf das Sprint Planning.....	157
Sprint Planning 1	158
Sprint Planning 2	162

11 Die Sprint-Durchführung 165

Artefakte und Ereignisse im Sprintablauf	165
Ausführen der Arbeit.....	167
Das Sprint Backlog	168
Das Scrum Taskboard.....	170

12	Das Daily Scrum	173
	Das Daily Scrum vorbereiten und durchführen.....	173
	Den Sprint-Umfang anpassen.....	177
	Einen Sprint abbrechen.....	179
	Den Fortschritt überwachen.....	180
	Das Taskboard.....	181
	Das Sprint Burndown Chart.....	181
	Das Release Burnup Chart.....	183
	Das Story Burndown Chart.....	184
	Der Themenpark.....	185
	Das Velocity Chart.....	186
13	Das Sprint Review	189
	Das Ziel des Sprint Reviews.....	189
	Vorbereitung und Ablauf des Sprint- Reviews.....	190
	Das Resultat des Sprint Reviews.....	193
14	Die Sprint Retrospektive	195
	Kontinuierliche Verbesserung und stetiges Lernen.....	195
	Vorbereiten der Retrospektive.....	197
	Die Retrospektive durchführen.....	199
	Sicherheit schaffen.....	199
	Ereignisse visualisieren.....	200
	Erkenntnisse sammeln.....	201
	Maßnahmen definieren.....	202
	Maßnahmen priorisieren.....	203
15	Zusammenarbeit im Scrum-Team	207
	Die neue Team-Philosophie.....	208
	Große Herausforderungen.....	210
	Fest zugeordnetes Team.....	210
	Limitierte Teamgröße.....	211
	Selbstorganisierende Teams.....	211
	Funktionsübergreifend arbeiten.....	215
	Kontinuierliche Verbesserung und Lernen.....	216

Ownership übernehmen	217
Die Zusammenarbeit im Team	218
Zusammenarbeit mit dem Product Owner	218
Zusammenarbeit mit dem Scrum Master	219
Zusammenarbeit mit dem Auftraggeber	222

16 Scrum bei großen Projekten 225

Große Projekte und verteilte Teams.....	225
Der agile Blueprint für große Projekte	228
Der Scrum@Scale™ Guide.....	229
Scrum@Scale Übersicht.....	229
Der Scrum Master Cycle.....	231
Der Product Owner Cycle.....	237
Die Organisation großer und verteilter Projekte.....	243
Die Praktiken für große Projekte anpassen.....	244
Das Product Backlog bei großen Projekten.....	244
Releaseplanung bei großen Projekten	245
Den Aufwand bestimmen.....	245
Projektweites Sprint Review und Retrospektive	246

17 Glossar 249

18 Anhang 261

Internet-Links und empfohlene Literatur	261
Über den Autor	263
Literaturverzeichnis	264
Stichwortverzeichnis.....	266



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 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 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“. 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 erfolgreich 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 Jahrzenten 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 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 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 44** 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.

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 (ohne Abbildungen) die Mindestanforderung an ein Scrum-Projekt. Sie können ihn hier kostenlos heruntergeladen:

<https://www.scrum.org/resources/scrum-guide>.

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

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.

Vorwort

1

Agiles Projektmanagement

Agile 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

Agiles Projektmanagement

und zu leben ist hingegen einiges schwieriger – und hier haben und werden viele Unternehmen noch Mühe haben.

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 19). 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:

Agiles Projektmanagement

- 1986 – Barry Boehm entwickelte erste Ansätze eines iterativen Software-Entwicklungsprozesses mit dem risikoorientierten Spiralmodell.
- 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, der Extreme Programming entwickelte, das Buch *Extreme Programming Explained* 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 25) 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.

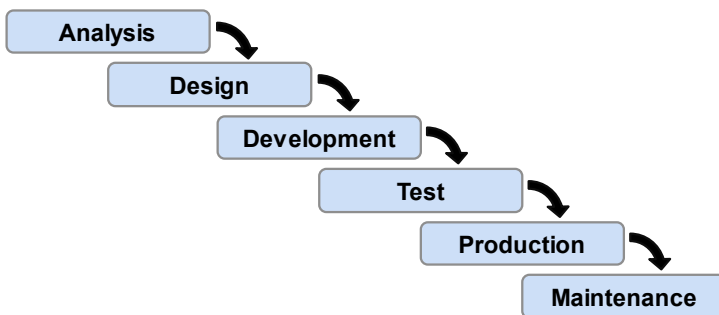


Abbildung 1: Das Wasserfall-Modell

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.

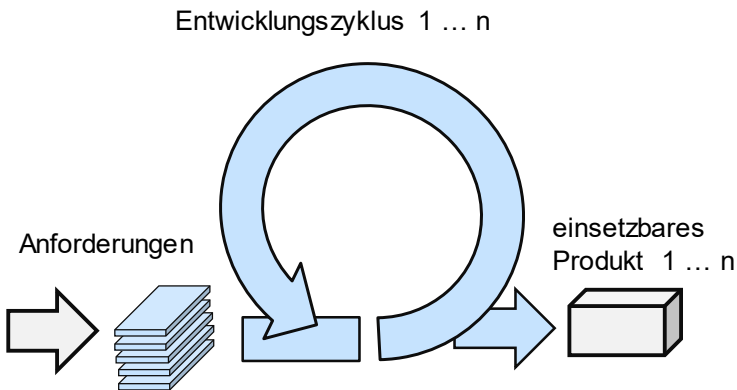


Abbildung 2: Agiles Vorgehen

Agiles Projektmanagement für alle Projekte?

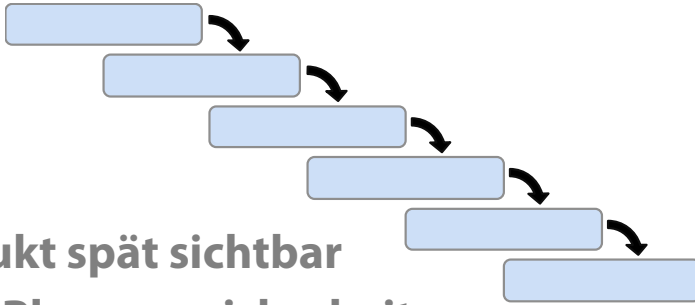
Agiles Projektmanagement setzt sich immer mehr durch, auch außerhalb der traditionellen Softwareentwicklung. Viele sagen dem Wasserfallmodell schon den Tod voraus. Soweit wird es aber nicht kommen. Versuchen Sie ein Haus, Flugzeugträger, Werkzeugmaschine oder eine Fabrikationsanlage mit agilem Projektmanagement 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.

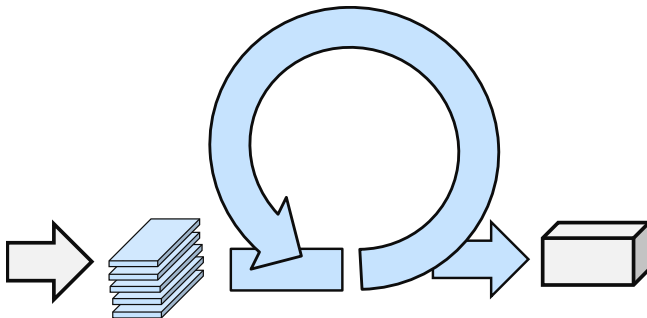
Bestimmte Prinzipien, Praktiken und Werte des agilen Projektmanagements werden sicher auch das traditionelle Projektmanagement in Zukunft stark beeinflussen. Ich denke nur schon an die Zusammenarbeit im Team und an die agilen Werte und Prinzipien. Aber überall wo physische Produkte entwickelt werden 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 Produkteentwicklung 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 Produkteentwicklungsteams. Der Inhalt des Artikels ist stark beeinflusst vom „Toyota Production System“ und japanischen Produkteentwicklungsmethoden 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 erfolgreicherem Produkten führt.
3. **Overlapping development phases:** Mit stark überlappenden Phasen wird der Entwicklungsprozess schneller und flexibler.

Glossar

Nachfolgend finden Sie eine Übersicht über die wichtigsten Begriffe von Scrum und dem agilen Projektmanagement, die ich in diesem Buch verwendet habe. Dies ist die erste Anlaufstelle, wenn Ihnen ein Begriff im agilen Projektmanagement unklar ist.

Agile Manifesto – Das im Jahr 2001 verabschiedete Manifest ist die Grundlage aller agilen Vorgehensweisen und beschreibt die zentralen agilen Werte.

Agile Prinzipien – Das Agile Manifesto beruht auf 12 Prinzipien, zu deren Einhaltung sich jeder Anwender agiler Methoden verpflichtet.

Agile Werte – Sind die wesentlichen Grundwerte für ein erfolgreiches, agiles Projektmanagement. Zusammengefasst: Mehr Flexibilität, weniger Strukturen, produktivere Zusammenarbeit. (siehe auch Agiles Manifesto)

Akzeptanzkriterien – Jedes Backlog Item (User Story) hat Akzeptanzkriterien. Diese sind Bestandteil der Fertigstellungskriterien (Definition of Done) und werden durch den Product Owner beim Review der fertigen User Story verwendet.

DEEP – Ein gute Product Backlog ist gemäß Mike Cohen DEEP. Dies bedeutet: Detailed, Estimated, Emergent (entwickelt sich), Prioritized.

Definition of Done (DoD) – Ist die Einigung eines agilen Teams darauf, was getan werden muss, damit ein Feature als fertig angesehen wird. Die DoD ist eine Liste von Fertigstellungskriterien.

Definition of Ready (DoR) – Ist eine Liste von Kriterien, die an die Product Backlog Items gestellt werden, damit diese bereit (Ready) sind, um im Sprint daran zu arbeiten. Im Gegensatz zur DoD ist der Product Owner für die Einhaltung der DoR verantwortlich.

DevOps – Durch die enge Zusammenarbeit von *Development and Operations* im agilen Umfeld und durch Automatisierung im ganzen Lebenszyklus können Releases schneller, in höherer Qualität, zu niedrigen Kosten und Risiken geliefert werden.

Dienende Führung – (Servant Leadership) Der Scrum Master ist dienender Führer für das Scrum-Team. Das heißt, er verfügt zwar über Einfluss, hat aber keine Autorität bezüglich der Arbeitsorganisation im Team. Er agiert als Coach und als Change Agent.

Dokumentation – Diese ist auch in agilen Projekten notwendig und hilfreich. Es gelten jedoch vorrangig die Grundsätze des Agilen Manifests (Working software over comprehensive documentation).

Done – In Scrum wird strikt unterscheiden zwischen »nicht fertig« und „fertig“. Ein Backlog Item gilt als „Done“, wenn alle Tasks erledigt sind und die Kriterien der Definition of Done erfüllt sind.

Development-Team – Das Development-Team führt alle Arbeiten aus, um die Anforderungen in ein auslieferbares Produktinkrement umzusetzen. Es organisiert seine Arbeit selbst.

Entwicklungsgeschwindigkeit (Velocity) – Ist die Summe aller Aufwände (Story Points), die das Scrum-Team in einem Sprint umgesetzt hat. Über die Zeit pendelt sich die Entwicklungsgeschwindigkeit auf einen relativ stabilen Wert ein. Die Entwicklungsgeschwindigkeit zu kennen ist für den Product Owner wichtig, damit er den Releaseplan erstellen und im diesen im Verlauf des Projektes justieren kann.

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

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
- Larsen, D. (2004). *Team Agility: Exploring Self-Organizing Software Development Teams*. Von <http://www.futureworksconsulting.com/resources/TeamAgilityAgileTimesFeb04.pdf> abgerufen

- McChrystal, G. S. (2015). In S. McChrystal, *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 107

A

Adaptive Software Development 31
Agil oder Lean 33
Agile Frameworks 228
Agile Manifest 12, **25**
Agile Manifests, Bedeutung 27
Agile Methoden 20, **31**
Agile Prinzipien **28**
Agile Prinzipien und Praktiken 34
Agilen Methoden, Unterschiede 32
Agiles Projektmanagement 15
Agiles Projektmanagement, Struktur **34**
Agilität im Projektmanagement 27
Agilität und Menschen 29
Agilität und Wissen 29
Agilität, was ist es 32
Aktivitäten definieren 163
Anforderung rangieren 102
Anforderungen 19, 45, 91, 99
Anforderungen erfassen 101
Anforderungen verstehen 102
Anforderungen, detaillierte 159
Anforderungen, gute 104
Anforderungen, High-Level 96
Anforderungen, nicht funktionale und technische **108**
Anforderungen, nicht funktionale, funktionale 96
Anforderungsbeschreibung 55
Anforderungsmanagement **91, 99**
Anforderungsworkshop 92, 101, 103, 113, 126
Anpassung 42
Anwender (User) **64**

Anwendererzählung 105
Arbeit ausführen 167
Arbeit priorisieren 167
Arbeit zuweisen 168
Arbeit, zu wenig/zu viel 177
Arbeitszyklen 45
Architektur und Design 162
Artefakt Transparency 82
Artefakte, weitere 47
Artifact Transparency 47
Auftraggeber **64**
Aufwand schätzen 117, 163
Aufwandschätzung **135**
Autoindustrie 10
Automatisierung 133

B

Backlog Refinement 83, 137
Barry Boehm 18
Burndown Chart 180, **181**
Burnup Charts 183
Business Case 49, 124
Business Process Engineering 10
Business Value **110**

C

Chief Product Owner **238**
Coach 62
CPO *Siehe* Chief Product Owner

D

Daily Scrum 77, **173**
Daily Scrum durchführen 174
Daily Scrum vorbereiten 173
Daily Scrum, Fragen 174
Daily Scrum, verteilte Teams 176
DEEP 115

Definition of Done 88, **155**
 Definition of Ready **153**
 Definitionsphase, trad. Projekt 99
 Development and Operations 131
 Development-Team **57**
 DevOps 131, 132
 Die Rollen in Scrum 46, **51**
 Distributed Teams 226
 Done 155

E

Empirie 41
 empirischer Prozesssteuerung 41
 Entwicklungsgeschwindigkeit 129,
 186
 Entwicklungsgeschwindigkeit
 bestimmen 147
 Epics **109**
 Ereignisse **73**
 Estimation Meeting 114, 116, **137**
 Estimation Meeting, Teilnehmer
 139
 Events **73**
 Executive Action Team (EAT) 235
Executive MetaScrum 241
 Explorationssprints 126
 externe Projekte 124
 Extreme Programming 31
 Extreme Programming (XP) 18

F

Feature Driven Development 31
 Feedback 192
 Fehlerkultur 72
 Feiern 194
 Fokus (Focus) **71**
 Fortschritt überwachen **180**
 Fortschrittsüberwachung 169
 Führungssysteme 15
 Funktionsübergreifend arbeiten 215
 funktionsübergreifende Teams 211

G

große Projekte 225
 Großunternehmen 15
 Grundursachen, finden 202

H

Herausforderungen, große 210
 Hindernisse 176

I

Impediment Backlog 204
 Impediments 176
 Initialisierungsphase 126
 Inspect & Adapt 46, 74, 77, 196
 INVEST, Kriterien 104
 Iterationen 45

J

japanischen
 Produkteentwicklungsmethoden
 23
 Jeff Sutherland 10, 25, 228

K

Karteikarten 105
 Ken Schwaber 10, 25, 228
 klassische Projekte **19**
 Kleinunternehmen 15
 Knowledge Management 24
 Kompensationssysteme 16, 208,
 221
 Konflikte im Team 220
 kontinuierliche Verbesserung 195
 Kontinuierliche Verbesserung 216
 kontinuierlichen
 Verbesserungsprozess 196
 Kunde **64**
 Kundenanforderungen verstehen
 102
 Kundenbedürfnisse 96

L

Large Scale Scrum 228
Lean Development 33
Lean Innovation 33
Lean Management 10, 24
Lean Production 10
leichtgewichtig 18
Leistungsbewertung 221
LeSS 228

M

Macht 16, 208
Magic Estimation 143
Manager **65**
Marktbedürfnisse 20
Märkte 96
Marktsegment 96
Mentor 62
Mentor und Coach 219
Minimum Marketable Feature Set
102, **110**
Minimum Viable Product **111**
Mittelfristplanung 128, 130
MoSCoW 121
Mut (Courage) **70**

N

Net Present Value 110
New New Product Development
Game 37

O

objektorientierte Programmierung
17
Offenheit (Openness) **71**
Ownership übernehmen 217

P

Parking Lot Diagram 185
Pflichten- und Lastenheft 124
Pflichtlieferobjekte 49

Planning Poker 142
Planung 128
Planung in Scrum **123**
Pre-Sprint Story Review 157
Priorisierung nach MoSCoW 121
Product Backlog 82, **113**
Product Backlog bei großen
Projekten 244
Product Backlog detaillieren 83
Product Backlog Management 55
Product Backlog priorisieren 120
Product Backlog, detaillieren 116
Product Backlog, Übersicht 114
Product Increment 85
Product Owner **54**
Product Owner Cycle 237
Product Owner, Zusammenarbeit
218
Product Vision Board 94
Product Vision, Team Identifikation
98
Product-Development-Teams 23
Produktattribute 96
Produkteentwicklungsteams 23
Produktlebenszyklen 17
Produktvision 55, **92**
Produktvision teilen,
kommunizieren 97
Projektbudget 124
Projekt-Governance 49
Projektorganisation, große Projekte
243
Projektplanung 123
Projektstart 124
Projekt-Startaktivitäten 124
Proof of Concept 127

R

Rapid Application Development
(RAD) 18
Rational Unified Process (RUP) 18
Ready Stories 159
Release-Management 130
Releaseplanung **128**

Releaseplanung bei großen Projekten 245
 Remote Teams 226
 Respekt **72**
 Retrospektive durchführen 199
 Retrospektive, Ablauf 198
 Retrospektive, Maßnahmen 202
 Retrospektive, Maßnahmen priorisieren 203
 Retrospektive, vorbereiten 197
 Return of Investment 110
 Return on Investment (ROI) 65
 Risiken 146
 Risiko der Anforderungen 118
 Risikoanalyse 124
 Rugby 24
 Rugby-Ansatz 37

S

SAFe® 211, 228
 ScALed 211, 228
 Scaled Agile Framework 228
 Scaled Agile Lean Development 228
 Scaled Daily Scrum 231, 232
 Schätzen mit Punkten 141
 Scope Änderungen 177
 Scrum als Management Framework 40
 Scrum Artefakte 47
 Scrum Artefakte – Übersicht 81
 Scrum bei großen Projekten 225
 Scrum But 48
 Scrum Ereignisse 46, **73**
 Scrum Framework **37**
 Scrum Guide 12, **38**
 Scrum im Schnellüberblick 44
 Scrum Master **58**
 Scrum Master Cycle 231
 Scrum Master Cycle, Ergebnisse 235
 Scrum Master, Aufgaben 58
 Scrum Master, Eigenschaften 62
 Scrum Master, Entscheidungen 62
 Scrum Master, Zusammenarbeit 219
 Scrum of Scrums 231
 Scrum of Scrums Master (SoSM) 233
 Scrum Taskboard 170
 Scrum Team, Umfeld 65
 Scrum Workflow **88**
 Scrum, die drei Säulen 41
 Scrum, Entstehung 37
 Scrum@Scale 229
 Scrum@Scale Guide 229
Scrums Master (SoSM) 232
 Scrum-Werte **67, 68**
 SDLC 17
 selbstorganisierende Teams 16
 Selbstorganisierende Teams **211**
 Selbstorganisierende und interdisziplinäre Teams 53
 Servant Leader 58, 219
 Set-Based Concurrent Engineering 33
 Six-Sigma 10
 Skalierung 234
 Softwareentwicklungsmethoden 17
 SoS skalieren 234
 Spiralmodell 18
 Sprint 74
 Sprint 0 126
 Sprint abbrechen 179
 Sprint Backlog 84, **168**
 Sprint Burndown Chart 181
 Sprint Planning 76, **149**
 Sprint Planning 1 **158**
 Sprint Planning 2 **162**
 Sprint Planning, Vorbereitung 157
 Sprint Retrospektive 80, **195**
 Sprint Review 78, **189**
 Sprint Review Teilnehmer 190
 Sprint Review, Ablauf 191
 Sprint Review, Maßnahmen daraus 193
 Sprint Review, Resultat 193
 Sprint-Ablauf 149
 Sprintdauer, optimale 146
 Sprint-Durchführung **165**

Anhang

Sprint-Review, Vorbereitung 191
Sprint-Reviews, Projektweite 246
Sprintstart, Wochentag 152
Sprint-Umfang anpassen 177
Sprintziel 158, 189
Stakeholder 51, 91, **189**, 192
Startaktivitäten 124
stetige Lernen 195
Story Burndown Chart 184
Systems Development Life Cycle
17

T

Taskboard **170**, 172, 180, 181
Taskboard, Informationszentrum
171
Task-Planung 167
Tasks **109**
Team, fest zugeordnetes 210
Teambildungsprozess, Tuckman
222
Teamgröße 147
Teamgröße, optimale 211
Teamkapazität überprüfen 164
Teamleistung und Belohnung 221
Team-Philosophie 208
Teams, mehrere 62
*The New New Product Development
Game* 23, 24, 209
Theme **109**
Themenpark 185
Timeboxing **74**
Timeline-Analyse 200
Time-to-Market 17
Toyota Production System 10, 23
traditionelle Projektmanagement 21

traditionellen Softwareentwicklung
21
Transparenz 42, 70, 71, 86
Transparenz der Artefakte 86

U

Überprüfung 42
Umfangänderungen 177
Unified Process 31
Unternehmens-Governance 49
Unternehmenskultur 15
User Stories 92
User Stories, kleine 145
User Story **105**

V

Velocity 147
Velocity Chart 186
Verpflichtung (Commitment) **69**
verteilte Teams 225
Vorprojekt 126

W

Wasserfallmodell 17, 21
Wasserfall-Modell 19
Wertesystem 27, 67
Wissensmanagement 10

Z

Zusammenarbeit im Scrum-Team
207
Zusammenarbeit in Teams 23
Zusammenarbeit mit Auftraggeber
222