

Ralf WIRDEMANN
Astrid RITSCHER
Johannes MAINUSCH



SCRUM mit User Stories

4. Auflage

HANSER

Wirdemann/Ritscher/Mainusch

Scrum mit User Stories



Blieben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

www.hanser-fachbuch.de/newsletter



Ralf Wirdemann
Astrid Ritscher
Johannes Mainusch

Scrum mit User Stories

4., überarbeitete und erweiterte Auflage

HANSER

Die Autor*innen:

Dipl.-Inform. Ralf Wirdemann, Hamburg
ralf.wirdemann@mailbox.org

Dipl.-Inform. Astrid Ritscher, Hamburg
astrid.ritscher@gmx.de

Dr. Johannes Mainusch, Hamburg
johannes.mainusch@kommitment.biz

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor:innen und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso übernehmen Autor:innen und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2022 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Brigitte Bauer-Schiewek

Copy editing: Petra Kienle, Fürstfeldbruck

Layout: le-tex publishing services GmbH

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Max Kostopoulos

Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Göttingen

Printed in Germany

Print-ISBN: 978-3-446-47369-0

E-Book-ISBN: 978-3-446-47438-3

E-Pub-ISBN: 978-3-446-47526-7

Inhalt

Vorwort zur 4. Auflage	XV
1 Einführung	1
1.1 Warum dieses Buch?	2
1.2 Struktur und Aufbau	3
1.3 Dankeschön	5
1.4 Feedback	6
2 Beispiel: Scrumcoaches.com	7
2.1 Das Projekt	8
2.2 Der Entwicklungsprozess	9
2.3 Die Beteiligten	10
2.4 Die Anforderungen	10
2.5 Priorisieren und Schätzen des Product Backlog	12
2.5.1 Priorisieren	12
2.5.2 Schätzen	13
2.6 Sprint-Planung	14
2.6.1 Sprint-Ziel	14
2.6.2 Entwicklungsgeschwindigkeit	15
2.6.3 Analyse der User Stories	15
2.6.4 Design der User Stories	16
2.7 Sprint-Durchführung	16
2.8 Messen des Sprint-Fortschritts	19
2.9 Am Ende des Sprint	20
2.9.1 Sprint-Review	20
2.9.2 Sprint-Retrospektive	21
2.10 Die Arbeit geht weiter	22
2.11 Zusammenfassung	23
2.12 Wie geht es weiter?	23

3	Die Grundlagen von Scrum	24
3.1	Was ist Scrum?	24
3.2	Scrum, ein Framework?	26
3.3	Überblick	27
3.3.1	Scrum-Team	27
3.3.2	Vision und Product Backlog	27
3.3.3	Sprint Planning Meeting	28
3.3.4	Sprints	29
3.3.5	Daily Scrums	29
3.3.6	Sprint-Review	29
3.3.7	Sprint-Retrospektive	30
3.4	Prinzipien	30
3.4.1	Transparenz	30
3.4.2	Beobachten und Anpassen	31
3.4.3	Timeboxing	31
3.4.4	Dinge abschließen	32
3.4.5	Maximierung von Geschäftswert	33
3.4.6	Teams scheitern nicht	34
3.4.7	Ergebnisorientierung	34
3.5	Die Rollen	35
3.5.1	Das Team	36
3.5.2	Der ScrumMaster	37
3.5.2.1	Dienstleistender Anführer und Problembeseitiger	37
3.5.2.2	Scrum implementieren	38
3.5.2.3	Entscheider	38
3.5.2.4	Müssen ScrumMaster programmieren können?	38
3.5.2.5	Product Owner-Coaching	39
3.5.2.6	Belastbare Persönlichkeit	39
3.5.2.7	Scrum in der Organisation verbreiten	39
3.5.3	Der Product Owner	40
3.5.3.1	Den Kunden repräsentieren	41
3.5.3.2	User Stories und Product Backlog	41
3.5.3.3	Mit dem Team durch den Sprint	42
3.5.3.4	Bestimmen, wann was fertig ist	42
3.5.4	Nebenrolle Kunde	42
3.6	Die ideale Arbeitsumgebung	44
3.7	Empirisches Management	44
3.8	Zusammenfassung	46
3.9	Wie geht es weiter?	47

4	User Stories	48
4.1	Was sind User Stories?	49
4.1.1	Story-Karte	50
4.1.2	Konversation	51
4.1.3	Akzeptanzkriterien	51
4.2	Warum User Stories?	52
4.3	User Stories schreiben	53
4.3.1	Die Sprache des Kunden	54
4.3.2	Benutzerrollen	54
4.3.3	User-Story-Muster	56
4.3.4	Epics	56
4.3.5	Themen	58
4.3.6	Wie viel Detail?	59
4.3.7	Keine Technik	60
4.3.8	Keine Benutzeroberfläche	60
4.4	Eigenschaften guter User Stories	60
4.4.1	Independent – unabhängige User Stories	60
4.4.2	Negotiable – verhandelbare User Stories	61
4.4.3	Valuable – wertvolle User Stories	61
4.4.4	Estimatable – schätzbare User Stories	62
4.4.5	Small – kleine User Stories	62
4.4.6	Testable – testbare User Stories	63
4.5	Zusammenfassung	64
4.6	Wie geht es weiter?	64
5	Agiles Schätzen	65
5.1	Was ist agiles Schätzen?	66
5.1.1	Relative Größe statt Dauer	66
5.1.2	Schätzen in Story Points	67
5.1.3	Wo bleibt die Dauer?	68
5.1.4	Argumentationshilfe für Story Points	68
5.2	Schätzen von User Stories	69
5.2.1	Größenordnungen und Punktsequenzen	70
5.2.2	Planungspoker	71
5.2.2.1	Schätzen im Team	74
5.2.2.2	Referenz-Story und Triangularisierung	74
5.2.2.3	Planungspoker funktioniert	76
5.2.3	Wann schätzen?	76
5.3	Zusammenfassung	77
5.4	Wie geht es weiter?	77

6	Agiles Planen	78
6.1	Was macht Planung agil?	78
6.2	Velocity	80
6.2.1	Tatsächliche Velocity	80
6.2.2	Angenommene Velocity	81
6.2.2.1	Angenommene Velocity = Tatsächliche Velocity	82
6.2.2.2	Mittlere Velocity	83
6.2.3	Velocity-basierte Planung	84
6.2.4	Nachhaltige Velocity	85
6.3	Agile Planung funktioniert	87
6.3.1	Velocity korrigiert Schätzfehler	87
6.3.2	Neubewertung von User Stories	88
6.3.3	Urlaub, Krankheit und ähnliche Ereignisse	89
6.3.4	Der Plan entsteht	89
6.4	Zusammenfassung	90
6.5	Wie geht es weiter?	90
7	User Stories fürs Product Backlog	91
7.1	Das Product Backlog	91
7.2	Das Product Backlog füllen	93
7.2.1	Anforderungswshops	95
7.2.2	Interviews, Markt-Feedback und Abstimmungsrunden	96
7.2.3	Überarbeitung und Pflege des Product Backlog	97
7.3	User Stories priorisieren	98
7.3.1	Finanzieller Wert	98
7.3.2	Kosten	99
7.3.3	Kundenzufriedenheit nach Kano	100
7.3.4	Risiko	101
7.3.5	Abhängigkeiten	102
7.3.6	Priorisierende Faktoren abwägen	102
7.3.7	MuSCoW-Priorisierung	103
7.4	User Stories schneiden	104
7.4.1	Vertikales Schneiden	104
7.4.2	Schneiden nach Daten	106
7.4.3	Schneiden nach Aufwand	106
7.4.4	Schneiden nach Forschungsanteilen	107
7.4.5	Schneiden nach Qualität	108
7.4.6	Schneiden nach Benutzerrolle	108

7.4.7	Schneiden nach Akzeptanzkriterien	109
7.4.8	Schneiden nach technischer Voraussetzung	110
7.5	Andere Anforderungen	110
7.5.1	Anforderungen umformulieren	111
7.5.2	Constraints	111
7.5.3	Fehler.....	112
7.5.4	Technisches Backlog	113
7.6	Zusammenfassung.....	114
7.7	Wie geht es weiter?.....	114
8	User Story Mapping	115
8.1	User Story Maps	116
8.2	Eine Story Map erstellen	117
8.2.1	Schritt 1: User Tasks ermitteln	118
8.2.2	Schritt 2: Gruppen bilden – User Activities.....	119
8.2.3	Schritt 3: Ordnung schaffen	119
8.2.4	Schritt 4: User Tasks durchlaufen = Geschichten erzählen	120
8.2.5	Schritt 5: User Stories schreiben	121
8.3	Warum Story Mapping?	122
8.3.1	Basis für gute Product Backlogs	122
8.3.2	Kleinstmögliche Releases.....	123
8.3.3	Motivation und Einsicht für alle Stakeholder	123
8.3.4	Lückenlosigkeit	123
8.3.5	Softwarearchitektur	123
8.3.6	Multi-Team-Setups	124
8.4	Von der Story Map zum Product Backlog.....	124
8.4.1	User Stories schreiben	126
8.4.2	Die Story Map ersetzt das Product Backlog	127
8.5	Zusammenfassung.....	127
8.6	Wie geht es weiter?.....	128
9	Sprint-Planung	129
9.1	Überblick und Ablauf.....	129
9.2	Beteiligte	130
9.3	Ergebnisse.....	130
9.4	Vorbereitung	133
9.4.1	Sprint Velocity.....	133
9.4.1.1	Anpassen der Velocity	133
9.4.1.2	Bugfixing, Refactoring und andere Aufgaben	134

9.4.2	Story-Auswahl.....	135
9.4.3	Sprint-Länge	135
9.5	Sprint Planning 1.....	137
9.5.1	Ablauf	137
9.5.2	Sprint-Ziel – warum führen wir den Sprint durch?.....	138
9.5.3	Vorstellung, Analyse und Commitment	138
9.5.4	Fehler und andere technische Aufgaben	140
9.6	Sprint Planning 2.....	141
9.6.1	Ablauf	142
9.6.2	Story-Design.....	142
9.6.3	Tasks schneiden.....	144
9.6.3.1	Taskgröße.....	145
9.6.3.2	Schneidetechniken.....	145
9.6.3.3	Ungeplante Tasks.....	146
9.6.4	Tasks schätzen?	146
9.6.4.1	Taskschätzungen sind sinnvoll	147
9.6.4.2	Taskschätzungen sind unsinnig	147
9.6.4.3	Keine Empfehlung	148
9.6.5	Das Sprint Backlog	149
9.6.6	Fehler und andere technischen Aufgaben verteilen	150
9.6.7	Was tun, wenn es länger wird?	150
9.7	Abschluss.....	151
9.8	Zusammenfassung.....	152
9.9	Wie geht es weiter?.....	152
10	Sprint-Durchführung	153
10.1	Die eigentliche Arbeit beginnt	153
10.2	Wer macht was?	155
10.2.1	Das Team	155
10.2.2	Der Product Owner.....	156
10.2.3	Der ScrumMaster	156
10.3	Story für Story Richtung Sprint-Ziel	157
10.3.1	Wie viele User Stories zurzeit?	158
10.3.2	Arbeit an einer User Story	158
10.3.3	Definition of Done	158
10.3.4	Abnahme fertiger User Stories	159
10.3.4.1	Entwicklertest.....	159

10.3.4.2	Akzeptanztest	160
10.3.4.3	QA-Abnahme	160
10.3.4.4	Frühestmögliche Fehlerbehebung	161
10.4	Daily Scrum	161
10.4.1	Aktualisierung des Taskboard	162
10.4.2	Ein guter Zeitpunkt	163
10.4.3	Ein guter Ort	164
10.4.4	Wer ist noch dabei?	164
10.4.5	Was macht der ScrumMaster?	165
10.5	Unterbrechungen	165
10.6	Messen und Anpassen	167
10.6.1	Bug- und technische Burndown-Charts	168
10.6.2	Was tun, wenn es eng wird?	168
10.7	Reguläres Sprint-Ende	170
10.8	Sprint-Review	171
10.8.1	Vorbereitung	171
10.8.2	Ort, Zeitpunkt und Teilnehmer	171
10.8.3	Ziel	171
10.8.4	Ablauf	172
10.9	Das Team organisiert sich	172
10.9.1	Verantwortung übernehmen	173
10.9.2	Das Team machen lassen und aus Fehlern lernen	173
10.9.3	Den Product Owner einbeziehen	174
10.9.4	Software-Pull-Systeme	174
10.9.5	Das Team bei der Arbeit mit Tasks coachen	175
10.9.6	Einzelgespräche	176
10.10	Sprint Best Practices	177
10.10.1	Source Code Management und Story-Branches	177
10.10.2	Kontinuierliches Integrieren	178
10.10.3	Automatisierung	178
10.10.4	Verständlicher Quellcode	178
10.10.5	Elektronische Sprint Backlogs und Burndown-Charts	179
10.11	Zusammenfassung	179
10.12	Wie geht es weiter?	180
11	User Stories Akzeptanztesten	181
11.1	Was ist Akzeptanztesten?	181

11.1.1	Akzeptanzkriterien	182
11.1.1.1	Akzeptanzkriterien sind Erwartungen	182
11.1.1.2	Akzeptanzkriterien sind Geschäftsregeln	183
11.1.2	Akzeptanztests	183
11.1.3	Akzeptanztesten	184
11.2	Akzeptanzkriterien schreiben	184
11.2.1	Vom Akzeptanzkriterium zum Akzeptanztest	185
11.2.2	Merkmale guter Akzeptanzkriterien	186
11.2.3	Akzeptanzkriterien auch für Epics?	187
11.3	Beispiel: Suche nach Coaches	188
11.4	Kleine Bausteine: Auf dem Weg zur DSL	189
11.5	Akzeptanztesten während des Sprint	190
11.6	Die hohe Schule: Akzeptanztest-getriebene Entwicklung	192
11.6.1	ATDD-Beispiel: Suche nach Coaches	193
11.6.2	Product Owner love writing Tests?	194
11.6.2.1	Alternative JCriteria	194
11.7	Lohnt sich das Ganze?	195
11.8	Zusammenfassung	196
11.9	Wie geht es weiter?	196
12	Sprint-Retrospektive	197
12.1	Nach dem Sprint ist vor dem Sprint	198
12.2	Ablauf von Retrospektiven	198
12.3	Retrospektiven vorbereiten	200
12.4	Retrospektiven leiten	200
12.5	Agenda und Check-in	201
12.6	Phase 1: Daten sammeln	202
12.6.1	Erstellung einer Timeline	203
12.6.2	Erweiterung der Timeline um Energiepunkte	204
12.7	Phase 2: Einsichten generieren	204
12.7.1	Positiv/Delta-Liste	205
12.7.2	Warum-Fragen	205
12.8	Phase 3: Entscheiden, was zu tun ist	206
12.9	Phase 4: Ziele formulieren und Aktionen planen	207
12.10	Abschluss	208
12.11	Themenorientierte Retrospektiven	208
12.12	Zusammenfassung	210
12.13	Wie geht es weiter?	210

13	Agile Releaseplanung	211
13.1	Releaseplanung	211
13.1.1	Themen-Releases	211
13.1.2	Datum-Releases	212
13.1.3	Releaseplanungs-Workshop	213
13.1.4	Was macht die Planung agil?	213
13.2	Planungs-Velocity	214
13.2.1	Durchführung von Test-Sprints	214
13.2.2	Historische Daten	214
13.2.3	Das Team bestimmen lassen	215
13.2.4	Auswahl eines Verfahrens	215
13.3	Der Releaseplan	216
13.4	Sichere Planung	217
13.4.1	Sichere Velocity	217
13.4.2	Sicherheit durch weniger wichtige User Stories	218
13.5	Monitoring und Aktualisierung	219
13.6	Zusammenfassung	220
13.7	Wie geht es weiter?	220
14	Mobiles Arbeiten	221
14.1	Herausforderungen	221
14.2	Start ins mobile Arbeiten	223
14.3	Mobiles Arbeiten in Scrum	224
14.3.1	Werkzeuge	225
14.3.1.1	Das digitale Whiteboard	225
14.3.1.2	Das digitale Taskboard	226
14.3.1.3	Mobiles Pair Programming	226
14.3.2	Meetings	226
14.3.2.1	Mobiler Start-Workshop	227
14.3.2.2	Mobiles Daily Scrum	227
14.3.2.3	Mobile Sprint-Planung	228
14.3.2.4	Mobile Retrospektive	228
14.3.2.5	Beispiel für ein Retrospektive Board	229
14.3.2.6	Mobiles Review	233
14.4	Die Zukunft: hybrides Arbeiten	233
14.4.1	Hybrid starten	234
14.4.2	Hybrid im Alltag	235
14.5	Zusammenfassung	235
14.6	Wie geht es weiter?	236

15	Verticals – SCRUM@OTTO	237
15.1	Warum ich über diese Geschichte schreibe	237
15.2	Die Vorgeschichte	239
15.3	Das Lhotse-Projekt – Zahlen, Daten, Fakten	240
15.4	Das Team – Menschen im Mittelpunkt.....	241
15.5	Triaden – die Führung eines Teams	243
15.6	Die Triade – Rollenbeschreibungen	243
15.6.1	Der Projektmanager – Project-Lead.....	244
15.6.2	Der Produktmanager – Business-Designer	244
15.6.3	Der Team-Architekt – Technical-Designer	245
15.7	Die TD-Runde	246
15.8	Die Otto-Architektur in Vertikalen.....	248
15.8.1	Warum die klassische IT versagt.....	248
15.8.2	Warum vertikale Schnitte helfen	251
15.8.3	Was eine Vertikale ist	252
15.8.4	Wie vertikale Schnitte gefunden werden können	253
15.9	Makro- und Mikroarchitektur	256
15.9.1	Makroarchitektur	256
15.9.2	Mikroarchitektur.....	257
15.10	Werte und Leitplanken statt Richtlinien und Governance	257
15.11	Das klassische Management in der agiler werdenden Organisation.....	258
15.12	Scrum@Otto – 100 Sprints später	259
15.13	Fazit	262
	Glossar	263
	Literatur	271
	Stichwortverzeichnis	273

Vorwort zur 4. Auflage

Wenn diese 4. Ausgabe von *Scrum mit User Stories* erscheint, hat sich nach mehr als zwei Jahren Pandemie ein Teil der Arbeitswelt stark verändert, insbesondere Arbeit, die sich dank Informationstechnologie vom Büro nach Hause verlagern lässt. Und es ist abzusehen, dass mobiles Arbeiten bleibt und hybride Arbeitsweisen, die es Teammitgliedern ermöglichen, von zu Hause oder aus dem Büro miteinander zu arbeiten, weiterentwickelt werden.

Softwareteams standen Anfang 2020 von heute auf morgen vor der Herausforderung, ihren Arbeitsalltag vom Büro ins Homeoffice zu verlegen. Plötzlich wurden Teams getrennt, für die es zuvor selbstverständlich war, im selben Raum und mit extrem kurzen und effizienten Kommunikationswegen zusammenzuarbeiten. Diese Zäsur der Arbeitswelt ist die Hauptmotivation für die Neuauflage dieses Buches: Nach mehr als zwei Jahren im Homeoffice hatten wir das Bedürfnis aufzuschreiben, was für uns funktioniert hat und worauf es beim mobilen Arbeiten ankommt.

Ich (Ralf) persönlich habe die Umstellung auf das Homeoffice in der Rolle des Entwicklers und Architekten miterlebt. Entsprechend war ich nicht hauptverantwortlich für die Neustrukturierung unserer neuen Arbeitssituation, sondern habe die Umstellung in mehreren Teams aus der zweiten Reihe mehr beobachtend und beratend begleitet. Aus diesem Grund war es mir wichtig, die bei der Mobilisierung agiler Teams entstehenden Herausforderungen aus erster Hand zu bekommen. Es wurde Zeit für eine neue Co-Autorin.

Für die 4. Auflage meines Buches konnte ich dafür Astrid Ritscher gewinnen. Astrid ist Agile-Coach bei der Otto GmbH & Co KG in Hamburg und stand vor gut zwei Jahren vor der Herausforderung, ihre agil arbeitenden Teams auf das mobile Arbeiten vorzubereiten. Astrid hat ihre Erfahrungen mit dem neuen Kapitel „Mobiles Arbeiten“ beigesteuert, in dem sie beschreibt, was sie und ihre Teams in den letzten zwei Jahren gelernt haben.

Das neue Kapitel widmet sich den Auswirkungen mobilen Arbeitens und seinen Anforderungen und Möglichkeiten an die Arbeit eines Teams und seiner Stakeholder mit Scrum. Für mobiles Arbeiten allgemein und die Besonderheiten von Scrum-Meetings und anderen agilen Arbeitsweisen (z. B. Pairing) werden Alternativen, Ideen und Empfehlungen vorgestellt. Der letzte Abschnitt des Kapitels geht auf hybrides Arbeiten ein.

Wir hoffen, Ihnen mit diesem neuen Thema eine weitere wichtige Facette der Agilen Softwareentwicklung näherzubringen, die Ihnen bei dem Neueinstieg oder der Optimierung Ihrer agilen Arbeitswelt behilflich ist.

Hinweis zur Verwendung geschlechterneutraler Sprache

Wenn bei personellen Bezeichnungen die männliche Form gewählt wurde (z. B. Mitarbeiter, Techniker), so sind damit in gleicher Weise weibliche Mitarbeiter oder Transgender-Mitarbeiter gemeint.

Hamburg, im Mai 2022

Ralf Wirdemann, Astrid Ritscher und Dr. Johannes Mainusch

1

Einführung

Das vorliegende Buch vereint zwei erfolgreiche Konzepte der Agilen Softwareentwicklung: Scrum und User Stories. Scrum ist ein Framework für das Management agiler Softwareprojekte. User Stories sind in der Sprache des Anwenders formulierte Anforderungen an das zu entwickelnde Softwaresystem. Ein in der Praxis erprobtes und häufig durchgeführtes Vorgehen ist die Verwendung von User Stories im Product Backlog als zentrales Werkzeug für die Planung und Priorisierung der Anforderungen eines Scrum-Projekts.

Anforderungen und Anforderungsmanagement sind die treibende Kraft in Software-Projekten. In Bezug auf Anforderungen beschränkt sich Scrum auf die Beschreibung, wie Anforderungen verwaltet werden, geht aber nicht darauf ein, woher sie kommen, wie man sie findet oder wie sie formuliert werden. Scrum ist ein sehr offenes Framework und macht bewusst keine Vorgaben über Art und Inhalt der Anforderungen im Product Backlog, dem zentralen Anforderungskatalog in Scrum. Dieses Buch optimiert Scrum hinsichtlich eines kundenorientierten Anforderungsmanagements, indem es die Verwendung von User Stories anstelle von „allgemeinen Items“ im Product Backlog vorschlägt. Dieser Ansatz ist sehr erfolgreich und in der Praxis weit verbreitet. Warum ist das so?

Schaltzentrale Product Owner

Entscheidend für die Beantwortung dieser Frage ist die Rolle des Product Owner, dem die Verwaltung des Product Backlog unterliegt. Der Product Owner eines Scrum-Projekts repräsentiert den Kunden und verfolgt dessen Interessen. Kunden denken geschäftsorientiert, das heißt, ein Kunde will wissen, was ihm die Umsetzung einer bestimmten Anforderung bringt. Und genau an diesem Punkt bieten User Stories einen entscheidenden Vorteil: Jede Story beschreibt einen greifbaren Mehrwert für den Kunden. Ein User-Story-basiertes Backlog macht es für den Product Owner viel einfacher, Entscheidungen zu treffen und Stories zu priorisieren, da er User Stories im Sinne einer wirtschaftlichen Kosten-Nutzen-Rechnung bewerten und einander gegenüberstellen kann.

Product Owner sind keine Techniker, das heißt, ein Product Owner kann im Regelfall mit der Anforderung „Refactoring der Datenbank-Zugriffsschicht“ wenig anfangen. Gelingt es uns aber, diese Refactoring-Aufgabe als Teil einer Mehrwert-erbringenden User Story zu formulieren, dann fällt es dem Product Owner wesentlich leichter, die Sinnhaftigkeit dieser Anforderung zu erkennen und sie entsprechend zu priorisieren.

Scrum mit User Stories kombinieren

User Stories spielen ihre Vorteile in unterschiedlichen Phasen und Elementen von Scrum aus: Product Backlogs enthalten sinnvolle, das heißt Geschäftswert-steigernde Produkt-erweiterungen. Die Sprint-Planung wird effektiver, da greifbarer Mehrwert analysiert und diskutiert wird. Eine an User Stories ausgerichtete Sprint-Durchführung ist zielgerichtet und lässt sich sinnvoll und mit Hilfe einfacher Regeln organisieren. Akzeptanztests orientieren sich an User Stories und geben vor, wann eine Story fertig und für den Anwender benutzbar ist.

Neben der Beschreibung, wie User Stories in den unterschiedlichen Phasen und Elementen von Scrum verwendet werden, ist das agile Schätzen und Planen mit Scrum und User Stories ein weiterer Schwerpunkt dieses Buches. Das Buch erklärt, wie ein Product Backlog in Story Points geschätzt wird, wie die Entwicklungsgeschwindigkeit eines Teams bestimmt und angepasst wird und wie sich basierend auf diesen Zahlen ein agiler Releaseplan sowie eine konkrete Sprint-Planung erstellen lassen.

Viele der Ideen in diesem Buch stammen von Ken Schwaber, Jeff Sutherland und Mike Beedle, den Begründern von Scrum, sowie von Mike Cohn, der für die zunehmende Popularität von User Stories verantwortlich ist. Was ich mit diesem Buch anbiete, ist eine Beschreibung, wie sich diese Ideen fortführen und effektiv kombinieren lassen und im Sinne eines hoch produktiven Softwareentwicklungsprozesses genutzt werden können.

■ 1.1 Warum dieses Buch?

User Stories sind eine in der Scrum-Gemeinschaft weit verbreitete Methode des Anforderungsmanagements. Egal, wohin Sie gucken, welches Forum Sie lesen oder in welches Scrum-Projekt Sie kommen, fast überall werden Anforderungen als User Stories beschrieben und im Product Backlog verwaltet. Schaut man sich hingegen die Literatur zum Thema an, dann findet man viele Bücher zu beiden Themen, aber bisher kein Buch, das die Themen kombiniert.

Ich habe während meiner Arbeit mit Scrum und User Stories gelernt, dass es viele Fragen gibt, die aus der Kombination dieser beiden Themen resultieren: Wie werden User Stories geschätzt und woher weiß ich, wie viele Stories in einen Sprint passen? Wie funktioniert eine effektive Sprint-Planung mit User Stories? Wie wird die Fehlerbehebung organisiert und geplant? Wie wirkt sich das auf die Entwicklungsgeschwindigkeit des Teams aus und muss ich das vorher berücksichtigen? Dies sind Beispiele für Fragen, auf die mir keines der existierenden Bücher Antworten liefern konnte. Entsprechend haben wir in den letzten Jahren sehr viel experimentiert und dazugelernt, indem wir uns für jeden Sprint etwas Neues überlegt und ausprobiert haben.

Sicher ist es so, dass jedes Team für sich und je nach Projektsituation seinen eigenen Weg finden muss. Dennoch denke ich, dass es eine Reihe bewährter Praktiken gibt, die wie in diesem Buch beschrieben oder entsprechend angepasst, auch in vielen anderen Projekten funktionieren und nicht von jedem Team mühsam aufs Neue erarbeitet werden müssen.

Dieses Buch beschreibt Scrum, so wie wir es benutzen, erhebt dabei aber keinen Anspruch auf Vollständigkeit oder darauf, den einzig richtigen Scrum-Weg zu beschreiben. Selbst wenn die beschriebenen Ansätze für Ihre Teams nicht immer funktionieren, so hege ich dennoch die Hoffnung, dass das Buch genügend Anreize und Impulse für die Entwicklung der für Sie funktionierenden Ideen liefert.

Die Arbeit mit User Stories hat mir darüber hinaus gezeigt, wie wichtig der Product Owner und dessen enge Zusammenarbeit mit dem Team ist. User Stories stellen den Product Owner in den Mittelpunkt des Geschehens und beziehen ihn eng in die Entwicklung und den Sprint mit ein. Dies ist ein Aspekt, den ich selbst erst lernen musste und deshalb in diesem Buch entsprechend darlegen will. Die Aufgabe des ScrumMaster ist es nicht nur, sich um das Team zu kümmern, sondern darüber hinaus, den Product Owner sehr stark in den Sprint mit einzubinden und dafür zu sorgen, dass Team und Product Owner eng zusammenarbeiten.

Das Schwierige an Scrum ist nicht, den Prozess zu verstehen, sondern ihn anzuwenden. Dieses Buch ist keine weitere theoretische Einführung in Scrum, sondern beschreibt einen praxiserprobten Ansatz. Es geht weniger um das „Was“ als um das „Wie“: Wie wende ich Scrum kombiniert mit User Stories erfolgreich an. „Scrum mit User Stories“ ist ein Handbuch für die Praxis, das beschreibt, wie wir Scrum einsetzen und wie die Kombination von Scrum und User Stories für uns funktioniert. Das Buch soll den Product Owner inspirieren, User Stories zu verwenden und im Rahmen von Scrum umzusetzen. Es soll CTOs motivieren, Scrum in zukünftigen Projekten auszuprobieren und Anforderungen mit Hilfe von User Stories zu beschreiben. Und das Buch soll dem Team dabei helfen, Spaß an der Arbeit zu haben, und durch die Verwendung von User Stories das Gefühl erzeugen, jeden Tag etwas Sinnvolles und Verwendbares zu produzieren.

■ 1.2 Struktur und Aufbau

Neben diesem Einführungskapitel enthält das Buch elf weitere Kapitel, die sich in vier logische Teile gliedern.

Teil 1: Grundlagen

Das Buch beginnt mit einem Grundlagenteil, bestehend aus einem Beispiel und einer allgemeinen Einführung in Scrum und User Stories. Das [Kapitel 2](#), *Beispiel Scrumcoaches.com*, ist ein Rundumschlag, der alle in diesem Buch behandelten Themen anhand eines Beispiels kurz und in sehr komprimierter Form anreißt. Das Kapitel liefert einen Überblick dessen, was Sie in diesem Buch erwartet, und steckt den Rahmen des Buches ab.

Neben dem Beispiel enthält der Grundlagenteil die [Kapitel 3](#), *Die Grundlagen von Scrum*, und [Kapitel 4](#), *User Stories*. Beide Kapitel führen die Themen *Scrum* und *User Stories* zunächst unabhängig voneinander ein und legen so das Fundament für die weiteren Kapitel dieses Buches.

Teil 2: Agiles Schätzen und Planen mit User Stories

Teil 2 des Buches liefert mit [Kapitel 5](#), *Agiles Schätzen*, und [Kapitel 6](#), *Agiles Planen*, eine Einführung in das Schätzen und Planen mit User Stories im Kontext von Scrum. Geschätzte User Stories sind die wesentliche Zutat für die Sprint- und Releaseplanung. Das Team entwickelt über die Zeit ein Gefühl dafür, wie viele User Stories welcher Größe es in einem Sprint umsetzen kann, und der Product Owner erstellt basierend auf den vom Team geschätzten User Stories eine über mehrere Sprints hinausgehende Releaseplanung.

Aufbauend auf den Schätz- und Planungs-Kapiteln spannt [Kapitel 7](#), *User Stories fürs Product Backlog*, den entscheidenden Bogen hin zur Kombination von Scrum und User Stories. Das Kapitel beschreibt, wie ein rein aus User Stories bestehendes Product Backlog erstellt wird und wie die Stories dafür priorisiert und auf die richtige Größe „geschnitten“ werden. Außerdem erklärt das Kapitel den Umgang mit Anforderungen, die sich nicht als User Stories beschreiben lassen.

Das in der 3. Auflage dieses Buches hinzugekommene [Kapitel 8](#) *User Story Mapping* beschreibt ein modernes Verfahren zur Anforderungsanalyse, bei dem von Anfang an ein ganzheitlicher Blick auf das Produkt eingenommen wird. Das Produkt wird dabei in ganzer Breite betrachtet und mit Hilfe von User Tasks visualisiert. Auf diese Art entsteht eine sinnstiftende Produktgeschichte, die verschiedensten Stakeholdern erzählt und mit ihnen gemeinsam fortgeschrieben wird. Als Ergebnis entsteht eine Story Map, als Basis für Releaseorientiert priorisierte Product Backlogs mit User Stories.

Teil 3: Sprint-Planung und -Durchführung

Das Herz und der Rhythmus von Scrum sind bestimmt durch Sprints, die eigentlichen Entwicklungsphasen in Scrum. Dieser Teil des Buches besteht aus vier Kapiteln, die sich mit der Planung und Durchführung von Sprints, dem Akzeptanztesten der im Sprint umgesetzten User Stories sowie dem Rückblick auf den Entwicklungsprozess beschäftigen. Das [Kapitel 9](#), *Sprint-Planung*, beschreibt den zu Beginn eines Sprint durchgeführten Planungsworkshop, in dem das Scrum-Team die anstehenden User Stories analysiert und deren Software-Design erstellt. Weiter geht es mit [Kapitel 10](#), *Sprint-Durchführung*, in dessen Mittelpunkt das Team und seine selbstorganisierte Arbeit stehen. Im Sprint arbeitet das Team in enger Zusammenarbeit mit dem Product Owner an der Umsetzung der geplanten User Stories und liefert zum Sprint-Ende getestete und funktionierende Software aus. Das Ergebnis des Sprint wird in einem öffentlichen Sprint-Review präsentiert, in dem alle Stakeholder zusammenkommen und Einfluss auf den weiteren Verlauf des Projekts nehmen.

Vor der abschließenden Sprint-Retrospektive führt [Kapitel 11](#), *User Stories Akzeptanztesten*, einen Exkurs zum Thema Akzeptanztesten durch. Akzeptanztesten ist weit mehr als eine reine Testdisziplin und beginnt bereits vor dem Sprint mit dem Schreiben von Akzeptanzkriterien und Akzeptanztests. Akzeptanzkriterien geben den roten Faden vor und zeigen dem Team, worauf es bei jeder Story ankommt. Akzeptanztests erklären die einzelnen Akzeptanzkriterien anhand konkreter Beispiele und bilden die Basis für den abschließenden Abnahmetest der User Stories.

Zum Abschluss jedes Sprint wirft das Team einen in [Kapitel 12](#), *Sprint-Retrospektive*, beschriebenen Rückblick auf seinen Entwicklungsprozess und analysiert, was gut war und was es in Zukunft zu verbessern gilt.

Teil 4: Releaseplanung

Der vorletzte Teil des Buches besteht aus einem einzigen Kapitel. Das [Kapitel 13](#), *Agile Releaseplanung*, beschreibt die Releaseplanung in Scrum mit Hilfe von User Stories. Ziel der Releaseplanung ist die Erstellung eines Releaseplans, der beschreibt, zu welchem Zeitpunkt welche User Stories voraussichtlich geliefert werden. Der Kunde kann sich mit Hilfe des Plans darauf einstellen, was er wann für sein Geld bekommt, und darüber hinaus entscheiden, ob sich das Projekt grundsätzlich für ihn lohnt. Für das Scrum-Team erfüllt ein Releaseplan darüber hinaus die Funktion eines Wegweisers, der die Vision des Projekts aufzeigt und einen über einzelne Sprints hinausgehenden Kontext definiert.

Teil 5: Mobiles Arbeiten

Bis zum März 2020 waren mobil arbeitende Softwareteams die Ausnahme. Diese Ausnahmebedingungen wurden von heute auf morgen zum Normalzustand. Diesem Thema widmet sich das in der 4. Auflage neu hinzugekommene [Kapitel 14](#), *Mobiles Arbeiten*. Das Kapitel beschreibt die Mobilisierung agil arbeitender Softwareteams und richtet den Fokus dabei speziell auf Scrum mit seinen Abläufen und Meetings.

Teil 6: Fallbeispiel

Im letzten Teil dieses Buches berichtet Johannes Mainusch von seinen Erfahrungen als Entwicklungsleiter beim Relaunch des E-Commerce-Shopsystems von Otto in Hamburg. In Hochzeiten arbeiteten mehr als 60 Entwickler, aufgeteilt auf bis zu zehn Teams in diesem Projekt. Die Größe des Projekts erforderte nicht nur eine bestimmte Organisationsform, sondern auch eine auf diese Organisationsform abgestimmte Softwarearchitektur. Johannes beschreibt in diesem Kapitel, wie Scrum bei Otto eingesetzt und an die speziellen Gegebenheiten angepasst wurde.

■ 1.3 Dankeschön

Viele Menschen haben mir beim Schreiben dieses Buches geholfen, wertvolles Feedback gegeben und mich über die lange Zeit des Schreibens hin motiviert. Diesen Menschen möchte ich danken. Durch euch ist dieses Buch viel besser geworden.

Ich danke meinen Reviewern Thomas Baustert, Steffen Gemkow, Boris Gloger, Bjarne Jansen, Bernd Oestereich, Uwe Petschke, Hubert Rosicka, Katja Roth, Dr. Gernot Starke, Sascha Teske, Hans-Martin Winkler, Henning Wolf und Henning Zuzmann.

Ich danke meinen Lektorinnen Margarete Metzger und Brigitte Bauer-Schiewek für die gute Zusammenarbeit und das vor langer Zeit in mich gesetzte Vertrauen, ein weiteres Scrum-Buch im Hanser-Verlag zu veröffentlichen. Darüber hinaus möchte ich mich bei Irene Weilhart für die freundliche, immer kompetente und prompte Hilfe bei allen Herstellungsfragen bedanken.

Ganz besonders möchte ich mich bei meiner Mutter Waltraud Wirdemann bedanken, die nicht nur während meiner Schulzeit meine Rechtschreibung korrigiert hat, sondern das heute immer noch und gerne tut.

Mein ganz spezieller Dank gilt meiner Frau Astrid, die viel mehr für mich getan hat, als das Cover dieses Buches zu gestalten und das Manuskript wieder und wieder zu lesen. Danke, dass du immer da bist.

■ 1.4 Feedback

Ich freue mich über Feedback jeglicher Art. Teilen Sie mir Ihre Hinweise, Korrekturen oder sonstigen Anmerkungen per E-Mail an ralf.wirdemann@mailbox.org mit. Vielen Dank!

2

Beispiel: Scrumcoaches.com

Dieses Kapitel beschreibt Scrum und User Stories am Beispiel der Entwicklung von *Scrumcoaches.com*, einer webbasierten Plattform für die Vermittlung von Dienstleistungen im Scrum-Umfeld. Das Beispiel streift alle wichtigen Phasen und Bestandteile von Scrum unter Einbeziehung von User Stories als Werkzeug fürs Anforderungsmanagement.

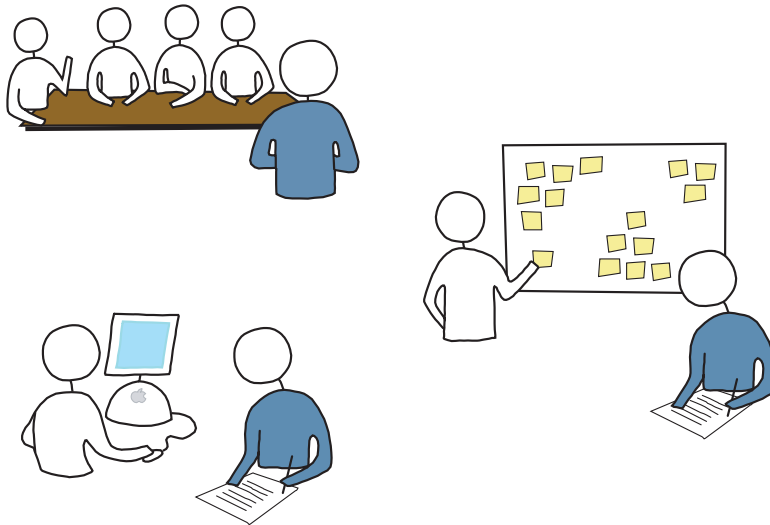


Abbildung 2.1 Das Scrumcoaches-Team bei der Arbeit

Das Kapitel begleitet das Scrumcoaches-Projektteam bei der Arbeit. Nach einer kurzen Einführung in den zugrunde liegenden Entwicklungsprozess und in die beteiligten Rollen werden die initiale Anforderungsanalyse im Rahmen eines Workshops und die Formulierung der resultierenden Anforderungen mit Hilfe von User Stories beschrieben, die User Stories geschätzt und hinsichtlich ihres Geschäftswerts priorisiert. Anschließend startet das Team in seine erste Iteration, in Scrum *Sprint* genannt. Im Sprint Planning Meeting, der Planungsphase eines Sprint, bespricht das Team eine Reihe von User Stories und zerbricht sie in konkrete Einzeltasks. In den kommenden zwei Wochen, der eigentlichen Entwicklungsphase des Sprint, arbeitet das Team die Tasks und damit die ausgewählten Stories nach und nach ab. Am Ende des Sprint liefert es funktionierende Software, die im Sprint-

Review allen interessierten Personen präsentiert wird. Den Abschluss des Sprint markiert die Sprint-Retrospektive, in der das Team seinen Entwicklungsprozess analysiert und Ideen für Verbesserungen entwickelt. Außerdem ermittelt das Team seine tatsächliche Entwicklungsgeschwindigkeit und kann für die folgenden Sprints besser einschätzen, wie viele User Stories in einen Sprint passen.

Das Kapitel ermöglicht es Ihnen, einen Schritt zurückzutreten und einen Gesamtblick auf ein Scrum-Projekt zu werfen. Ausgestattet mit diesem Überblick, geht es in den folgenden Kapiteln in die Details von Scrum und User Stories, in denen die hier nur kurz und exemplarisch vorgestellten Konzepte aufgegriffen und ausführlich beschrieben werden.

■ 2.1 Das Projekt

Scrum-Projekte beginnen mit einer Vision, welche die zentrale Idee des Vorhabens auf den Punkt bringt:

Wir wollen eine Web-Anwendung entwickeln, mit der Scrum-Coaches neue Projekte und Projektanbieter qualifizierte Scrum-Coaches finden können.

Basierend auf dieser Vision erstellt der Product Owner ein Konzeptpapier, das die Kernidee und Basisfunktionalität von Scrumcoaches.com zusammenfasst. Der folgende Abschnitt enthält einen Auszug aus diesem Konzeptpapier.

Produktkonzept: Scrumcoaches.com

Scrumcoaches.com soll eine webbasierte Plattform für die Vermittlung von Dienstleistungen im Scrum-Umfeld werden. Scrum-Coaches können ihr Profil hinterlegen und die Projektdatenbank nach offenen Projekten durchsuchen. Projektanbieter können Projekte einstellen und für die Suche freigeben oder gezielt nach Scrum-Coaches suchen. Beide Parteien können über die Plattform Kontakt zueinander aufnehmen. Darüber hinaus bietet das System eine Bewertungsfunktion, mit der Projektanbieter die Arbeit von Scrum-Coaches bewerten und diesen zu einer guten Reputation verhelfen können.

Ein wesentliches Merkmal von Scrumcoaches.com ist die Vermittlung von qualitativ hochwertigem Scrum-Know-how. Scrumcoaches.com-Kunden sollen sich sicher sein, dass die registrierten Coaches über umfangreiche Erfahrung aus mehreren Scrum-Projekten verfügen. Die Schlüsselemente hierfür sind Reputation und Vertrauen. Ein registrierter Scrum-Coach erlangt oder steigert seine Reputation, indem er sich von ehemaligen Kunden oder Teams bewerten lässt. Coaches geben also nicht nur selber ihre Fähigkeiten an, sondern übertragen diese Aufgabe an diejenigen, die bereits mit dem Coach zusammengearbeitet haben. Darüber hinaus kann ein Scrum-Coach seine eigene Reputation auf andere Coaches übertragen, indem er diese weiterempfiehlt. Coaches stehen dann mit ihrem Namen für andere Coaches ein.

Als Refinanzierung von Scrumcoaches.com ist derzeit eine abgestufte Rechnungsstellung angedacht. Coaches sollen die Anwendung in ihrer Basisversion kostenlos nutzen können. Der Weg für kostenpflichtige Dienste für Coaches soll jedoch offen gehalten werden. Projektanbieter zahlen bei erfolgreicher Vermittlung und für das Einstellen von Anzeigen.

■ 2.2 Der Entwicklungsprozess

Scrumcoaches.com wird mit Scrum entwickelt, einem Framework für das Management agiler Softwareprojekte. Scrum ist kein Entwicklungsprozess oder Vorgehensmodell im herkömmlichen Sinne, in welchem von Anfang an klar definiert ist, wann und in welcher Reihenfolge welche Ergebnisse zu liefern sind. Stattdessen beschreibt Scrum ausschließlich einen Rahmen, in dem sich das Entwicklungsteam und andere Interessenvertreter des Projekts relativ frei bewegen und arbeiten können. Dieser Abschnitt gibt einen kurzen Überblick über die wesentlichen Bestandteile, Rollen und Artefakte von Scrum.

Alle Anforderungen eines Scrum-Projekts werden in einem zentralen Anforderungskatalog, dem *Product Backlog*, verwaltet. Das Product Backlog enthält eine Liste von User Stories. Eine User Story beschreibt eine konkrete Funktionalität aus Sicht des Anwenders.

Scrum-Entwicklungsteams arbeiten in Sprints. Ein Sprint ist eine Entwicklungsperiode von ein bis vier Wochen Länge. In einem Sprint setzt das Team selbstorganisiert und eigenverantwortlich einen Teil der User Stories des Product Backlog in funktionierende Software um.

Scrum beschreibt einen iterativen Prozess der Softwareentwicklung, bestehend aus den folgenden vier Phasen:

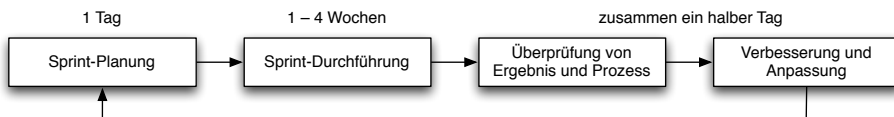


Abbildung 2.2 Die vier Phasen von Scrum

In der Sprint-Planung werden die User Stories aus dem Product Backlog ausgewählt, die im anstehenden Sprint umgesetzt werden sollen. In der Sprint-Durchführung werden die ausgewählten User Stories vom Team programmiert und ins Gesamtsystem integriert. Am Ende des Sprint werden das Ergebnis vom Kunden und der Entwicklungsprozess vom Team überprüft. Die daraus resultierenden Verbesserungsvorschläge werden im folgenden Sprint berücksichtigt und umgesetzt. Der beschriebene Kreislauf setzt sich über die gesamte Laufzeit des Projekts fort.

■ 2.3 Die Beteiligten

An der Entwicklung von Scrumcoaches.com sind drei Rollen beteiligt: der Product Owner, das Team und der ScrumMaster.

Der *Product Owner* repräsentiert den Kunden und somit den Anforderer. Er verfügt über das notwendige Domänenwissen, um Anforderungen zu formulieren und Detailfragen beantworten zu können. Der Product Owner ist der fachliche Projektleiter und für den Gesamterfolg des Projekts verantwortlich.

Das *Team* besteht aus drei Softwareentwicklern und einem HTML-Designer. Es ist für die erfolgreiche Umsetzung der User Stories eines Sprint verantwortlich. Das Team arbeitet selbstorganisiert im geschützten Raum des Sprint. Es gibt keinen Projektleiter, der dem Team sagt, was es zu tun oder zu lassen hat.

Der *ScrumMaster*¹ ist für die Einführung und Umsetzung von Scrum zuständig. Er coacht das Team, gibt aber keine Handlungsanweisungen oder weist Aufgaben zu. Ein ScrumMaster ist kein Vorgesetzter. Ein ScrumMaster räumt Hindernisse aus dem Weg und sorgt für optimale Arbeitsbedingungen. Der ScrumMaster hält dem Team den Rücken frei.

Neben diesen drei Hauptrollen gibt es in Scrum-Projekten auch immer die Rolle des Kunden, das heißt desjenigen, der die Software in Auftrag gibt und nach ihrer Entwicklung betreibt, nutzt oder verkauft. Im Fall von Scrumcoaches.com ist der Kunde ein junges Start-up-Unternehmen, dessen Geschäftsidee die beschriebene Plattform ist. Das Unternehmen besteht aus den beiden Gründern, die die Software bei einem auf Scrum spezialisierten Softwarehaus in Auftrag geben.

■ 2.4 Die Anforderungen

Anforderungen werden in Scrum im zentralen Product Backlog verwaltet. Das Product Backlog enthält eine Liste mit priorisierten und geschätzten User Stories. Eine User Story ist eine in der Sprache des Kunden beschriebene Anforderung an das System, die einen konkreten und für den Kunden sichtbaren Mehrwert liefert. User Stories werden in einem aktiven Stil geschrieben, der in ein bis zwei Sätzen den Kern der Anforderung auf den Punkt bringt. Ein Beispiel:

Als Scrum-Coach will ich nach Projekten suchen.

Für das Schreiben der User Stories und deren Verwaltung im Product Backlog ist der Product Owner zuständig. Nur er darf neue Stories hinzufügen, die Priorität vorhandener Stories ändern oder auch mit der Zeit unwichtig gewordene Stories aus dem Backlog entfernen.

¹ Für die Rolle des ScrumMaster hat sich die Camel-Case-Schreibweise etabliert. Leider konnte ich keinen Hinweis darauf finden, warum das so ist. Auch Ken Schwaber verwendet in seinem Original-Scrum-Buch noch die getrennte Schreibweise „Scrum Master“ [Schwaber und Beedle, 2001], stellt dann aber in seinen weiteren Büchern kommentarlos auf „ScrumMaster“ um [Schwaber 2004].

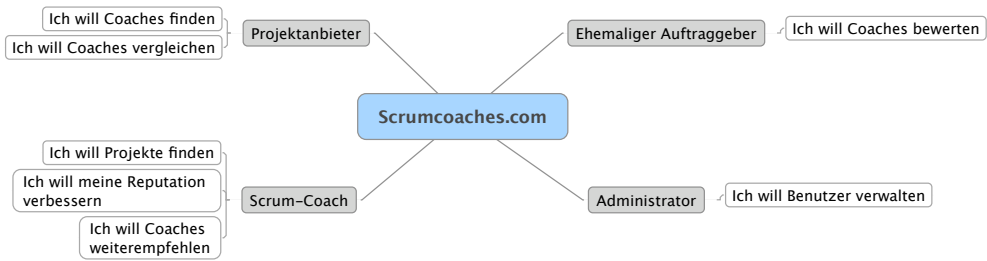


Abbildung 2.3 Mindmap mit den Zielen einzelner Benutzerrollen

Natürlich kann der Product Owner das Product Backlog nicht ganz alleine füllen. Dafür gibt es zu viele wichtige Ideen anderer Interessenvertreter, die der Product Owner sammeln und als User Stories formuliert ins Backlog übertragen muss. Deshalb führt der Product Owner zu Beginn des Projekts einen Anforderungsworkshop mit allen an Scrumcoaches.com interessierten Personen durch.

Gemeinsam wird eine Mindmap mit den zentralen Ideen und Inhalten der Plattform erstellt. Dazu schlüpft das Anforderungsteam in die verschiedenen Benutzerrollen (graue Kästchen in [Abbildung 2.3](#)) und überlegt sich, was Scrum-Coaches und Projektanbieter, aber auch ehemalige Auftraggeber oder Administratoren mit dem System erreichen wollen.

Im Anschluss an den Workshop zieht sich der Product Owner einige Stunden zurück und formuliert auf Basis der erstellten Mindmap eine Reihe von User Stories und trägt sie ins Product Backlog ein ([Tabelle 2.1](#)).

Tabelle 2.1 Scrumcoaches.com – erstes Product Backlog

User Story	
Als Scrum-Coach will ich nach Projekten suchen.	!
Als Scrum-Coach will ich mich auf Projekte bewerben.	!
Als Scrum-Coach will ich mein Profil einstellen.	!!!
Als Scrum-Coach will ich andere Coaches empfehlen.	!
Als Scrum-Coach will ich mich an- und abmelden.	!!!
Als Scrum-Coach will ich mich registrieren.	!!!
Als Projektanbieter will ich nach Coaches suchen.	!!!
Als Projektanbieter will ich eine Anfrage an einen Coach stellen.	!!!
Als Projektanbieter will ich mich registrieren.	!!!
Als Projektanbieter will ich mich an- und abmelden.	!!!
Als Projektanbieter will ich Projekte einstellen.	!
Als ehemaliger Kunde will ich Coaches bewerten.	!

■ 2.5 Priorisieren und Schätzen des Product Backlog

Das Product Backlog ist ein dynamisches Dokument. Im Laufe des Projekts ändern sich Anforderungen, es kommen neue hinzu, Prioritäten ändern sich oder Anforderungen fallen ganz weg. Außerdem nimmt der Detailgrad einer User Story zu, je näher die Story an ihre konkrete Umsetzung, das heißt an den nächsten Sprint rückt. Und genau dies ist die nächste Aufgabe des Product Owner. Er muss entscheiden, welche Stories im Product Backlog am wichtigsten sind, und sie entsprechend priorisieren.

2.5.1 Priorisieren

Bei seinen Erwägungen, welche User Stories am wichtigsten sind, überlegt sich der Product Owner als Erstes, was die Anwendung minimal leisten muss, damit sie in einer Basisversion nutzbar ist. Er beschließt, dass das System in einer Minimalversion potenziellen Projektanbietern Zugriff auf die Profile der registrierten Scrum-Coaches bieten sowie eine Funktion zur Kontaktaufnahme mit den Coaches bereitstellen muss. Ausgehend von diesen Überlegungen identifiziert der Product Owner eine Reihe von „Must Have“²-Stories, das heißt User Stories, ohne die das System nicht funktionieren würde.

Im Sinne der beschlossenen Minimalfunktion sind dies die Registrierungs-, An- und Abmeldestories für Coaches und Projektanbieter, die Profileinstellungs-Story für Coaches sowie die Stories zur Coach-Suche und anschließenden Kontaktaufnahme. Der Product Owner markiert die „Must Have“-Stories mit drei Ausrufezeichen. Ohne diese Stories geht es nicht. Sind diese Stories umgesetzt, dann bietet das System einen Mehrwert für Anbieter, die Coaches aktiv finden können, aber auch schon für Scrum-Coaches, die zwar noch nicht aktiv suchen, aber bereits kontaktiert werden können.

Als Nächstes bestimmt der Product Owner die „Should Have“-Stories und markiert sie mit einem Ausrufezeichen. Dies sind die User Stories, ohne die das System zwar funktionieren würde, die aber einen solch großen Mehrwert liefern, dass sie neben den „Must Haves“ zusätzlich umgesetzt werden sollten. Zwei gute Beispiele für „Should Have“-Stories sind die Bewertungs- und Empfehlungsstory. Das System funktioniert auch ohne sie, aber schließlich geht es um die Vermittlung von qualifizierten Coaches, und diese beiden Stories bilden die Grundlage für den Qualifikationsnachweis der registrierten Coaches. Weitere „Should Have“-Stories sind die Projekterfassungs-Story sowie die Story zur Projektsuche und Bewerbung für Coaches.

Der Product Owner sortiert die Stories im Product Backlog nach „Must Have“ und „Should Have“. Zusätzlich bringt er die Stories innerhalb ihrer Prioritätsklasse in eine aus entwicklungstechnischer Sicht sinnvolle Reihenfolge. Beispielsweise ist das An- und Abmelden ohne vorherige Registrierung nicht möglich. Es besteht eine Abhängigkeit zwischen beiden

² „Must Have“ ist eines von vier MuSCoW-Kriterien zur Priorisierung von User Stories. Ich verwende in diesem Beispiel nur zwei der Kriterien und beschreibe die vollständige MuSCoW-Priorisierung in [Kapitel 7, User Stories fürs Product Backlog](#).