

Inhaltsverzeichnis

1 Einführung	1
1.1 Warum dieses Buch?	2
1.2 Struktur und Aufbau	3
1.3 Dankeschön	5
1.4 Feedback	5
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	11
2.5.1 Priorisieren	11
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	17
2.8 Messen des Sprint-Fortschritts	19
2.9 Am Ende des Sprint	20
2.9.1 Sprint-Review	21
2.9.2 Sprint-Retrospektive	21
2.10 Die Arbeit geht weiter	22
2.11 Zusammenfassung	23
2.12 Wie geht es weiter?	24

3 Die Grundlagen von Scrum	25
3.1 Was ist Scrum?	26
3.2 Scrum, ein Framework?	27
3.3 Überblick	28
3.3.1 Scrum-Team	28
3.3.2 Vision und Product Backlog	29
3.3.3 Sprint Planning Meeting	30
3.3.4 Sprints	30
3.3.5 Daily Scrums	31
3.3.6 Sprint-Review	31
3.3.7 Sprint-Retrospektive	31
3.4 Prinzipien	32
3.4.1 Transparenz	32
3.4.2 Beobachten und Anpassen	32
3.4.3 Timeboxing	33
3.4.4 Dinge abschließen	34
3.4.5 Maximierung von Geschäftswert	35
3.4.6 Teams scheitern nicht	36
3.4.7 Ergebnisorientierung	36
3.5 Die Rollen	36
3.5.1 Das Team	37
3.5.2 Der ScrumMaster	39
3.5.3 Der Product Owner	43
3.5.4 Nebenrolle Kunde	45
3.6 Die ideale Arbeitsumgebung	46
3.7 Empirisches Management	47
3.8 Zusammenfassung	49
3.9 Wie geht es weiter?	50
4 User Stories	51
4.1 Was sind User Stories?	52
4.1.1 Story-Karte	53
4.1.2 Konversation	54
4.1.3 Akzeptanzkriterien	54
4.2 Warum User Stories?	55
4.3 User Stories schreiben	57
4.3.1 Die Sprache des Kunden	57

4.3.2	Benutzerrollen	58
4.3.3	User-Story-Muster	59
4.3.4	Epics	60
4.3.5	Themen	62
4.3.6	Wie viel Detail?	62
4.3.7	Keine Technik	63
4.3.8	Keine Benutzeroberfläche	63
4.4	Eigenschaften guter User Stories	64
4.4.1	Independent – Unabhängige User Stories	64
4.4.2	Negotiable – Verhandelbare User Stories	65
4.4.3	Valuable – Wertvolle User Stories	65
4.4.4	Estimatable – Schätzbare User Stories	66
4.4.5	Small – Kleine User Stories	66
4.4.6	Testable – Testbare User Stories	67
4.5	Zusammenfassung	68
4.6	Wie geht es weiter?	68
5	Agiles Schätzen	69
5.1	Was ist agiles Schätzen?	70
5.1.1	Relative Größe statt Dauer	70
5.1.2	Schätzen in Story Points	71
5.1.3	Wo bleibt die Dauer?	72
5.1.4	Argumentationshilfe für Story Points	72
5.2	Schätzen von User Stories	73
5.2.1	Größenordnungen und Punktesequenzen	74
5.2.2	Planungspoker	76
5.2.3	Wann schätzen?	80
5.3	Zusammenfassung	81
5.4	Wie geht es weiter?	81
6	Agiles Planen	83
6.1	Was macht Planung agil?	83
6.2	Velocity	85
6.2.1	Tatsächliche Velocity	85
6.2.2	Angenommene Velocity	87
6.2.3	Velocity-basierte Planung	89
6.2.4	Nachhaltige Velocity	91
6.3	Agile Planung funktioniert	93

6.3.1	Velocity korrigiert Schätzfehler	93
6.3.2	Neubewertung von User Stories	93
6.3.3	Urlaub, Krankheit und ähnliche Ereignisse	94
6.3.4	Der Plan entsteht	95
6.4	Zusammenfassung	95
6.5	Wie geht es weiter?	96
7	User Stories fürs Product Backlog	97
7.1	Das Product Backlog	98
7.2	Das Product Backlog füllen	100
7.2.1	Anforderungsworkshops	101
7.2.2	Interviews, Markt-Feedback und Abstimmungsrunden	103
7.2.3	Überarbeitung und Pflege des Product Backlog	103
7.3	User Stories priorisieren	104
7.3.1	Finanzieller Wert	104
7.3.2	Kosten	105
7.3.3	Kundenzufriedenheit nach Kano	106
7.3.4	Risiko	108
7.3.5	Abhängigkeiten	109
7.3.6	Priorisierende Faktoren abwägen	109
7.3.7	MuSCoW-Priorisierung	110
7.4	User Stories schneiden	111
7.4.1	Vertikales Schneiden	111
7.4.2	Schneiden nach Daten	113
7.4.3	Schneiden nach Aufwand	113
7.4.4	Schneiden nach Forschungsanteilen	114
7.4.5	Schneiden nach Qualität	115
7.4.6	Schneiden nach Benutzerrolle	116
7.4.7	Schneiden nach Akzeptanzkriterien	116
7.4.8	Schneiden nach technischer Voraussetzung	117
7.5	Andere Anforderungen	118
7.5.1	Anforderungen umformulieren	118
7.5.2	Constraints	119
7.5.3	Fehler	120
7.5.4	Technisches Backlog	121
7.6	Zusammenfassung	121
7.7	Wie geht es weiter?	122

8 Sprint-Planung	123
8.1 Überblick und Ablauf	123
8.2 Beteiligte	124
8.3 Ergebnisse	125
8.4 Vorbereitung	127
8.4.1 Sprint Velocity	127
8.4.2 Story-Auswahl	129
8.4.3 Sprint-Länge	130
8.5 Sprint Planning 1	131
8.5.1 Ablauf	131
8.5.2 Sprint-Ziel – Warum führen wir den Sprint durch?	132
8.5.3 Vorstellung, Analyse und Commitment	133
8.5.4 Fehler und andere technische Aufgaben	135
8.6 Sprint Planning 2	136
8.6.1 Ablauf	137
8.6.2 Story-Design	137
8.6.3 Tasks schneiden	139
8.6.4 Tasks schätzen?	141
8.6.5 Das Sprint Backlog	144
8.6.6 Fehler und andere technischen Aufgaben verteilen	145
8.6.7 Was tun, wenn es länger wird?	145
8.7 Abschluss	146
8.8 Zusammenfassung	147
8.9 Wie geht es weiter?	148
 9 Sprint-Durchführung	149
9.1 Die eigentliche Arbeit beginnt	149
9.2 Wer macht was?	151
9.2.1 Das Team	151
9.2.2 Der Product Owner	152
9.2.3 Der ScrumMaster	152
9.3 Story für Story Richtung Sprint-Ziel	153
9.3.1 Wie viele User Stories zurzeit?	154
9.3.2 Arbeit an einer User Story	154
9.3.3 Definition of Done	155
9.3.4 Abnahme fertiger User Stories	156
9.4 Daily Scrum	158

9.4.1	Aktualisierung des Taskboard	159
9.4.2	Ein guter Zeitpunkt	160
9.4.3	Ein guter Ort	160
9.4.4	Wer ist noch dabei?	161
9.4.5	Was macht der ScrumMaster?	161
9.5	Unterbrechungen	162
9.6	Messen und Anpassen	164
9.6.1	Bug- und technische Burndown-Charts	164
9.6.2	Was tun, wenn es eng wird?	165
9.7	Reguläres Sprint-Ende	167
9.8	Sprint-Review	168
9.8.1	Vorbereitung	168
9.8.2	Ort, Zeitpunkt und Teilnehmer	168
9.8.3	Ziel	168
9.8.4	Ablauf	169
9.9	Das Team organisiert sich	170
9.9.1	Verantwortung übernehmen	170
9.9.2	Das Team machen lassen und aus Fehlern lernen	171
9.9.3	Den Product Owner einbeziehen	171
9.9.4	Software-Pull-Systeme	172
9.9.5	Das Team bei der Arbeit mit Tasks coachen	173
9.9.6	Einzelgespräche	173
9.10	Sprint Best Practices	174
9.10.1	Source Code Management und Story-Branches	174
9.10.2	Kontinuierliches Integrieren	175
9.10.3	Automatisierung	176
9.10.4	Verständlicher Quellcode	176
9.10.5	Elektronische Sprint Backlogs und Burndown-Charts	176
9.11	Zusammenfassung	177
9.12	Wie geht es weiter?	178
10	User Stories Akzeptanztesten	179
10.1	Was ist Akzeptanztesten?	179
10.1.1	Akzeptanzkriterien	180
10.1.2	Akzeptanztests	181
10.1.3	Akzeptanztesten	182
10.2	Akzeptanzkriterien schreiben	183

10.2.1 Vom Akzeptanzkriterium zum Akzeptanztest	184
10.2.2 Merkmale guter Akzeptanzkriterien	185
10.2.3 Akzeptanzkriterien auch für Epics?	186
10.3 Beispiel: Suche nach Coaches	186
10.4 Kleine Bausteine: Auf dem Weg zur DSL	188
10.5 Akzeptanztesten während des Sprint	189
10.6 Die hohe Schule: Akzeptanztest-getriebene Entwicklung	191
10.6.1 ATDD-Beispiel: Suche nach Coaches	192
10.6.2 Product Owner love writing Tests?	193
10.7 Lohnt sich das Ganze?	194
10.8 Zusammenfassung	195
10.9 Wie geht es weiter?	196
11 Sprint-Retrospektive	197
11.1 Nach dem Sprint ist vor dem Sprint	198
11.2 Ablauf von Retrospektiven	198
11.3 Retrospektiven vorbereiten	200
11.4 Retrospektiven leiten	201
11.5 Agenda und Check-in	201
11.6 Phase 1: Daten sammeln	203
11.6.1 Erstellung einer Timeline	203
11.6.2 Erweiterung der Timeline um Energiepunkte	205
11.7 Phase 2: Einsichten generieren	205
11.7.1 Positiv- / Delta-Liste	205
11.7.2 Warum-Fragen	206
11.8 Phase 3: Entscheiden, was zu tun ist	207
11.9 Phase 4: Ziele formulieren und Aktionen planen	208
11.10 Abschluss	209
11.11 Themenorientierte Retrospektiven	209
11.12 Zusammenfassung	211
11.13 Wie geht es weiter?	211
12 Agile Releaseplanung	213
12.1 Releaseplanung	213
12.1.1 Themen-Releases	213
12.1.2 Datum-Releases	214
12.1.3 Releaseplanungs-Workshop	215
12.1.4 Was macht die Planung agil?	216

12.2 Planungs-Velocity	216
12.2.1 Durchführung von Test-Sprints	216
12.2.2 Historische Daten	217
12.2.3 Das Team bestimmen lassen	217
12.2.4 Auswahl eines Verfahrens	217
12.3 Der Releaseplan	218
12.4 Sichere Planung	219
12.4.1 Sichere Velocity	220
12.4.2 Sicherheit durch weniger wichtige User Stories	221
12.5 Monitoring und Aktualisierung	221
12.6 Zusammenfassung	222
12.7 Wie geht es weiter?	223
Glossar	225
Literaturverzeichnis	233
Stichwortverzeichnis	237