

Inhaltsverzeichnis

Zitate – Scrum	1
Vorwort von Mike Cohn	19
Vorwort von Ron Jeffries	21
Einleitung	23
Danksagungen	27
Über den Autor	31
I Einführung	33
I.1 Was ist Scrum?	33
I.2 Die Ursprünge von Scrum	35
I.3 Wieso Scrum?	36
I.4 Ergebnisse bei Genomica	37
I.5 Kann Scrum Ihnen helfen?	37
I.5.1 Die Complex-Domäne	40
I.5.2 Die Complicated-Domäne	40
I.5.3 Die Simple-Domäne	41
I.5.4 Die Chaotic-Domäne	41
I.5.5 Disorder (Nicht-Wissen, Regellosigkeit)	41
I.5.6 Unterbrechungsgesteuerte Arbeit	42
I.6 Abschließende Bemerkungen	43
Teil I Kernkonzepte	45
2 Das Scrum-Framework	47
2.1 Überblick	47
2.2 Scrum-Rollen	48
2.2.1 Product Owner	49
2.2.2 ScrumMaster	49

2.2.3	Das Entwicklungsteam	50
2.3	Scrum-Aktivitäten und Artefakte	50
2.3.1	Product Backlog	52
2.3.2	Sprints	54
2.3.3	Sprint-Planung	55
2.3.4	Sprint-Ausführung	56
2.3.5	Daily Scrum.	57
2.3.6	Fertig (Done)	58
2.3.7	Sprint Review	60
2.3.8	Sprint-Retrospektive	61
2.4	Abschließende Bemerkungen	61
3	Agile Prinzipien	63
3.1	Überblick	63
3.2	Veränderlichkeit und Unsicherheit	66
3.2.1	Hilfreiche Veränderlichkeit bereitwillig annehmen	66
3.2.2	Iterative und inkrementelle Entwicklung nutzen.	67
3.2.3	Ausnutzen der Veränderlichkeit durch Inspektion, Anpassung und Transparenz.	69
3.2.4	Gleichzeitiges Reduzieren aller Formen der Unsicherheit	70
3.3	Vorhersage und Anpassung	71
3.3.1	Optionen offen halten	71
3.3.2	Akzeptieren, dass man es nicht gleich von Anfang an richtig machen kann	72
3.3.3	Einen adaptiven, untersuchenden Ansatz bevorzugen	74
3.3.4	Änderung auf eine ökonomisch sinnvolle Weise annehmen	75
3.3.5	Vorhersagende, im Voraus erfolgende Arbeit mit adaptiver, bedarfsgerechter Arbeit abwägen.	78
3.4	Validiertes Wissen	79
3.4.1	Schnelles Validieren wichtiger Annahmen	79
3.4.2	Abwägen mehrerer gleichzeitiger Lernschleifen	79
3.4.3	Organisieren des Workflows für schnelle Feedbacks	80
3.5	Work in Process (WIP)	82
3.5.1	Wirtschaftlich sinnvolle Batch-Größen benutzen	82
3.5.2	Lagerbestände erkennen und sinnvoll verwalten.	84
3.5.3	Auf unerledigte Arbeit konzentrieren, nicht auf untätige Arbeiter.	85
3.5.4	Verzögerungskosten betrachten	87

3.6	Fortschritt	88
3.6.1	An Echtzeitinformationen anpassen und umplanen	88
3.6.2	Fortschritt messen, indem man funktionierende Güter validiert	89
3.6.3	Auf eine wertzentrierte Auslieferung konzentrieren	89
3.7	Leistung	90
3.7.1	Gehe schnell, aber hetze nicht	90
3.7.2	Baue Qualität ein	91
3.7.3	Mache alles ohne großes Zeremoniell	91
3.8	Abschließende Bemerkungen	92
4	Sprints	95
4.1	Überblick	95
4.2	Timeboxing	96
4.2.1	Legt ein WIP-Limit fest	96
4.2.2	Erzwingt eine Priorisierung	96
4.2.3	Demonstriert Fortschritt	97
4.2.4	Verhindert unnötigen Perfektionismus	97
4.2.5	Motiviert die Fertigstellung	97
4.2.6	Verbessert die Vorhersagbarkeit	98
4.3	Kurze Zeitdauer	98
4.3.1	Erleichterte Planung	98
4.3.2	Schnelles Feedback	99
4.3.3	Verbesserter Return on Investment	99
4.3.4	Begrenzte Fehler	99
4.3.5	Wiedererweckte Begeisterung	99
4.3.6	Häufige Checkpoints	100
4.4	Konsistente Dauer	101
4.4.1	Vorteile der Kadenz	102
4.4.2	Vereinfacht die Planung	102
4.5	Keine das Ziel beeinflussenden Änderungen	103
4.5.1	Was ist ein Sprint-Ziel?	103
4.5.2	Gegenseitige Verpflichtung	103
4.5.3	Änderungen versus Klärung	104
4.5.4	Konsequenzen einer Änderung	104
4.5.5	Pragmatisch sein	106
4.5.6	Abnormaler Abbruch	107
4.6	Definition von Fertig (Done)	108
4.6.1	Wie lautet die Definition von Fertig?	108

4.6.2	Die Definition von Fertig kann sich im Laufe der Zeit weiterentwickeln	110
4.6.3	Definition von Fertig versus Akzeptanzkriterien	112
4.6.4	Fertig versus Fertig-Fertig	112
4.7	Abschließende Bemerkungen	113
5	Anforderungen und User Stories	115
5.1	Überblick	115
5.2	Gespräche	117
5.3	Progressive Verfeinerung	118
5.4	Was sind User Stories?	119
5.4.1	Card (Karte)	119
5.4.2	Conversation (Gespräch)	120
5.4.3	Confirmation (Bestätigung)	121
5.5	Der Detaillierungsgrad	122
5.6	In gute Stories INVESTieren	125
5.6.1	Independent (unabhängig)	125
5.6.2	Negotiable (verhandelbar)	126
5.6.3	Valuable (werthaltig)	126
5.6.4	Estimatable (schätzbar)	128
5.6.5	Passende Größe (klein)	128
5.6.6	Testable (prüfbar)	129
5.7	Nichtfunktionale Anforderungen	129
5.8	Stories zum Wissenserwerb	130
5.9	Stories sammeln	132
5.9.1	Workshop zum Schreiben von User Stories	132
5.9.2	Story Mapping	133
5.10	Abschließende Bemerkungen	134
6	Das Product Backlog	137
6.1	Überblick	137
6.2	Product-Backlog-Elemente	138
6.3	Merkmale guter Product Backlogs	139
6.3.1	Detailed appropriately (ausreichend detailliert)	139
6.3.2	Emergent	140
6.3.3	Estimated (geschätzt)	140
6.3.4	Prioritized (priorisiert)	141
6.4	Pflege	142
6.4.1	Was bedeutet Pflege?	142

6.4.2	Wer führt die Pflege durch?	143
6.4.3	Wann findet die Pflege statt?	144
6.5	Die Definition von Bereit	146
6.6	Flow Management	148
6.6.1	Release Flow Management	148
6.6.2	Sprint Flow Management	149
6.7	Welche und wie viele Product Backlogs?	150
6.7.1	Was ist ein Produkt?	150
6.7.2	Große Produkte – hierarchische Backlogs	152
6.7.3	Mehrere Teams – ein Product Backlog	153
6.7.4	Ein Team – mehrere Produkte	154
6.8	Abschließende Bemerkungen	155
7	Schätzung und Velocity	157
7.1	Überblick	157
7.2	Was und wann wir schätzen	158
7.2.1	Schätzungen für Portfolio-Backlog-Elemente	159
7.2.2	Product-Backlog-Schätzungen	159
7.2.3	Aufgabenschätzungen	160
7.3	Schätzkonzepte für Product-Backlog-Elemente	161
7.3.1	Als Team schätzen	161
7.3.2	Schätzungen sind keine Verpflichtungen	162
7.3.3	Exaktheit versus Präzision	163
7.3.4	Relative Größenschätzung	163
7.4	Schätzeinheiten für Product-Backlog-Elemente	166
7.4.1	Story-Punkte	166
7.4.2	Idealtage	166
7.5	Planungspoker	167
7.5.1	Schätzskala	168
7.5.2	Wie man spielt	169
7.5.3	Vorteile	171
7.6	Was ist Velocity?	171
7.7	Einen Velocity-Bereich berechnen	172
7.8	Die Velocity vorhersagen	173
7.9	Die Velocity beeinflussen	174
7.10	Missbrauch der Velocity	175
7.11	Abschließende Bemerkungen	176

8	Technische Schulden	177
8.1	Überblick	177
8.2	Die Folgen technischer Schulden.....	179
8.2.1	Unvorhersehbarer Wendepunkt	180
8.2.2	Zunehmend verzögerte Auslieferung.....	180
8.2.3	Beträchtliche Anzahl an Defekten.....	180
8.2.4	Steigende Entwicklungs- und Support-Kosten.....	180
8.2.5	Das Produkt verkümmert.....	181
8.2.6	Schwindende Vorhersehbarkeit.....	181
8.2.7	Leistungseinbruch	181
8.2.8	Allgemeiner Frust.....	182
8.2.9	Sinkende Kundenzufriedenheit.....	182
8.3	Ursachen der technischen Schulden	182
8.3.1	Druck hinsichtlich des Erreichens einer Deadline.....	182
8.3.2	Erfolglose Versuche der Velocity-Beschleunigung.....	183
8.3.3	Gerücht: Weniger Testen kann die Velocity beschleunigen.....	184
8.3.4	Schulden bauen auf Schulden auf.....	185
8.4	Technische Schulden müssen organisiert werden	186
8.5	Die Zunahme technischer Schulden überwachen	187
8.5.1	Bewährte technische Praktiken anwenden	187
8.5.2	Eine starke Definition von Fertig benutzen	188
8.5.3	Die wirtschaftlichen Aspekte technischer Schulden richtig verstehen	188
8.6	Technische Schulden sichtbar machen	191
8.6.1	Technische Schulden auf geschäftlicher Ebene sichtbar machen	191
8.6.2	Technische Schulden auf der technischen Ebene sichtbar machen	192
8.7	Technische Schulden abbauen	194
8.7.1	Nicht alle technischen Schulden sollten abgebaut werden	195
8.7.2	Wenden Sie die Pfadfinderregel an (Bauen Sie die Schulden ab, sobald sie Ihnen begegnen).....	197
8.7.3	Bauen Sie technische Schulden schrittweise ab.....	197
8.7.4	Bauen Sie die technischen Schulden mit den höchsten Zinsen zuerst ab.....	198

8.7.5	Technische Schulden abbauen, während man für den Kunden werthaltige Arbeit erledigt	199
8.8	Abschließende Bemerkungen	200
Teil II Rollen		201
9	Der Product Owner	203
9.1	Überblick	203
9.2	Hauptaufgaben	204
9.2.1	Organisation der wirtschaftlichen Belange	205
9.2.2	Mitwirkung an der Planung	207
9.2.3	Pflege des Product Backlogs	207
9.2.4	Definition und Verifikation der Akzeptanzkriterien	207
9.2.5	Zusammenarbeit mit dem Entwicklungsteam	208
9.2.6	Zusammenarbeit mit den Stakeholdern.....	209
9.3	Eigenschaften/Fähigkeiten	210
9.3.1	Fachwissen	210
9.3.2	Soziale Kompetenz.....	211
9.3.3	Entscheidungsfindung.....	211
9.3.4	Verantwortung.....	211
9.4	Der Alltag eines Product Owners.....	212
9.5	Wer sollte Product Owner sein?	214
9.5.1	Interne Entwicklung	215
9.5.2	Gewerbliche Entwicklung	216
9.5.3	Ausgelagerte Entwicklung	218
9.5.4	Komponentenentwicklung	219
9.6	Product Owner kombiniert mit anderen Rollen	220
9.7	Das Product-Owner-Team.....	220
9.7.1	Product-Owner-Stellvertreter.....	221
9.7.2	Chief Product Owner	222
9.8	Abschließende Bemerkungen	223
10	ScrumMaster	225
10.1	Überblick	225
10.2	Wichtigste Aufgaben	225
10.2.1	Coach.....	225
10.2.2	»Dienende Führungskraft«	226

10.2.3	Prozessautorität	227
10.2.4	Schutz vor störenden Einflüssen	227
10.2.5	Beseitigung von Hindernissen	227
10.2.6	Berater in der Organisationsentwicklung	227
10.3	Eigenschaften/Fähigkeiten	228
10.3.1	Sachkundig	228
10.3.2	Neugierig	228
10.3.3	Geduldig	229
10.3.4	Teamfähig	229
10.3.5	Schützend	229
10.3.6	Transparent	230
10.4	Alltag	230
10.5	Die Rolle ausfüllen	231
10.5.1	Wer sollte ScrumMaster sein?	231
10.5.2	Ist die Rolle des ScrumMasters eine Vollzeitbeschäftigung?	232
10.5.3	ScrumMaster in Kombination mit anderen Rollen	232
10.6	Abschließende Bemerkungen	233
II	Das Entwicklungsteam	235
II.1	Überblick	235
II.2	Rollenspezifische Teams	235
II.3	Wichtigste Aufgaben	236
II.3.1	Durchführung des Sprints	236
II.3.2	Tägliches Untersuchen und Anpassen (»Inspect and Adapt«)	237
II.3.3	Pflege des Product Backlogs	237
II.3.4	Den Sprint planen	237
II.3.5	Produkt und Prozess untersuchen und anpassen	237
II.4	Eigenschaften/Fertigkeiten	238
II.4.1	Selbstorganisierend	238
II.4.2	Funktionsübergreifend vielseitig	240
II.4.3	T-förmige Fertigkeiten	241
II.4.4	Die Musketier-Einstellung	243
II.4.5	Kommunikation mit hoher Bandbreite	245
II.4.6	Transparente Kommunikation	246
II.4.7	Die richtige Größe	246
II.4.8	Fokussiert und verpflichtet	247
II.4.9	In einer nachhaltigen Geschwindigkeit arbeiten	249

11.5	11.4.10 Langlebig	250
	Abschließende Bemerkungen	252
12	12 Die Strukturen des Scrum-Teams	253
12.1	12.1 Überblick	253
12.2	12.2 Funktionsteams versus Komponententeams	253
12.3	12.3 Koordination mehrerer Teams	258
	12.3.1 Scrum of Scrums	258
	12.3.2 Der Release-Train	260
12.4	12.4 Abschließende Bemerkungen	263
13	13 Manager	265
13.1	13.1 Überblick	265
13.2	13.2 Teams koordinieren	266
	13.2.1 Grenzen definieren	267
	13.2.2 Ein klares Ziel vorgeben	267
	13.2.3 Teams bilden	268
	13.2.4 Die Teamzusammensetzung ändern	269
	13.2.5 Teams bevollmächtigen	269
13.3	13.3 Teams fördern	271
	13.3.1 Die Mitarbeiter anspornen	271
	13.3.2 Kompetenz entwickeln	271
	13.3.3 Fachliche Anleitung bieten	272
	13.3.4 Die Integrität des Teams bewahren	273
13.4	13.4 Die Umgebung ausrichten und anpassen	273
	13.4.1 Agile Werte fördern	273
	13.4.2 Organisatorische Hindernisse entfernen	274
	13.4.3 Die internen Abteilungen ausrichten	274
	13.4.4 Die Partner ausrichten	275
13.5	13.5 Den Wertschöpfungsfluss organisieren	275
	13.5.1 Die Sichtweise des Systems annehmen	275
	13.5.2 Die wirtschaftlichen Aspekte organisieren	276
	13.5.3 Messungen und Berichte überwachen	276
13.6	13.6 Projektmanager	277
	13.6.1 Projektmanagementaufgaben in einem Scrum-Team	277
	13.6.2 Eine getrennte Projektmanager-Rolle bewahren	279
13.7	13.7 Abschließende Bemerkungen	283

Teil III Planen	285
14 Scrum-Planungsprinzipien	287
14.1 Überblick	287
14.2 Gehen Sie nicht davon aus, dass im Voraus erstellte Pläne korrekt sind	288
14.3 Die Vorabplanung sollte hilfreich, aber nicht exzessiv sein	288
14.4 Halten Sie sich die Planungsoptionen bis zum letzten verantwortbaren Augenblick offen	289
14.5 Konzentrieren Sie sich stärker auf das Anpassen und Neuplanen als darauf, einem Plan zu genügen	289
14.6 Den Planungsbestand richtig organisieren	292
14.7 Bevorzugen Sie kleinere und häufigere Releases	293
14.8 Lernen Sie schnell dazu und weichen Sie vom Plan ab, wenn es nötig sein sollte	295
14.9 Abschließende Bemerkungen	295
15 Planung auf mehreren Ebenen	297
15.1 Überblick	297
15.2 Portfolio-Planung	299
15.3 Produktplanung (Visionsbildung)	299
15.3.1 Die Vision	299
15.3.2 Allgemeines Product Backlog	300
15.3.3 Produkt-Roadmap	300
15.4 Release-Planung	302
15.5 Sprint-Planung	304
15.6 Tägliche Planung	304
15.7 Abschließende Bemerkungen	305
16 Portfolio-Planung	307
16.1 Überblick	307
16.1.1 Das Timing	307
16.1.2 Teilnehmer	308
16.1.3 Der Prozess	308
16.2 Zeitplanungsstrategien	310
16.2.1 Optimierung der Rendite über die Lebensdauer	311
16.2.2 Kalkulation der Verzögerungskosten	312
16.2.3 Schätzungen mit Genauigkeit statt Präzision	315
16.3 Zuflussstrategien	316
16.3.1 Den wirtschaftlichen Filter anwenden	316

16.3.2	Zufluss- und Abflussrate ausbalancieren	317
16.3.3	Sich bietende Gelegenheiten schnell ergreifen	319
16.3.4	Planen Sie kleinere, häufigere Releases	320
16.4	Abflussstrategien	322
16.4.1	Auf unerledigte Arbeit konzentrieren, nicht auf untätige Mitarbeiter	322
16.4.2	Einrichten eines WIP-Limits	322
16.4.3	Auf ein komplettes Team warten	323
16.5	Strategien zur Überprüfung der in Bearbeitung befindlichen Produkte	324
16.5.1	Die Grenznutzenrechnung verwenden	325
16.6	Abschließende Bemerkungen	326
17	Visionsfindung (Produktplanung)	329
17.1	Überblick	329
17.1.1	Das Timing	330
17.1.2	Teilnehmer	330
17.1.3	Der Prozess	332
17.2	SR4U-Beispiel	332
17.3	Die Entwicklung der Vision	334
17.4	Erstellen eines allgemeinen Product Backlogs	336
17.5	Die Definition der Produkt-Roadmap	337
17.6	Andere Aktivitäten	340
17.7	Wirtschaftlich vernünftige Visionsfindung	342
17.7.1	Eine realistische Vertrauensschwelle anstreben	343
17.7.2	Konzentrieren Sie sich auf einen kurzfristigen Planungshorizont	344
17.7.3	Handeln Sie schnell	345
17.7.4	Erwerben Sie validiertes Wissen	345
17.7.5	Nutzen Sie eine inkrementelle Finanzierung	346
17.7.6	Lernen Sie schnell dazu und weichen Sie ggf. vom Plan ab (aka Schnelles Scheitern)	348
17.8	Abschließende Bemerkungen	348
18	Release-Planung (längerfristige Planung)	349
18.1	Überblick	349
18.1.1	Das Timing	350
18.1.2	Teilnehmer	350
18.1.3	Der Prozess	351

18.2	Release-Einschränkungen	353
18.2.1	Alles fest	353
18.2.2	Umfang und Termin fest	354
18.2.3	Fester Umfang	355
18.2.4	Fester Termin	356
18.2.5	Variable Qualität	356
18.2.6	Einschränkungen aktualisieren	356
18.3	Das Product Backlog pflegen	357
18.4	Die minimal freigebbaren Funktionen (Minimum Releasable Features, MRFs) verfeinern	358
18.5	Sprint Mapping (Einordnung der Product-Backlog-Elemente)	359
18.6	Release-Planung mit festem Termin	361
18.7	Release-Planung mit festem Umfang	366
18.8	Die Kosten berechnen	368
18.9	Kommunizieren	369
18.9.1	Den Fortschritt in einem Release mit festem Umfang kommunizieren	369
18.9.2	Den Fortschritt in einem Release mit festem Termin kommunizieren	372
18.10	Abschließende Bemerkungen	373
Teil IV Sprints		375
19	Die Sprint-Planung	377
19.1	Überblick	377
19.1.1	Das Timing	377
19.1.2	Teilnehmer	377
19.1.3	Der Prozess	378
19.2	Ansätze für die Sprint-Planung	380
19.2.1	Zweiteilige Sprint-Planung	380
19.2.2	Einteilige Sprint-Planung	381
19.3	Die Kapazität ermitteln	382
19.3.1	Was ist die Kapazität?	382
19.3.2	Kapazität in Story-Punkten	384
19.3.3	Die Kapazität in Aufwandsstunden	384
19.4	Product-Backlog-Elemente auswählen	385
19.5	Zuversicht erwerben	386

19.6	Das Sprint-Ziel verfeinern	388
19.7	Die Verpflichtung finalisieren	388
19.8	Abschließende Bemerkungen	388
20	Die Sprint-Ausführung	389
20.1	Überblick	389
20.1.1	Das Timing	389
20.1.2	Teilnehmer	390
20.1.3	Der Prozess	390
20.2	Die Planung der Sprint-Ausführung	391
20.3	Flow-Management	391
20.3.1	Parallele Arbeit und Ausschwärmen	392
20.3.2	Welche Arbeit begonnen werden soll	394
20.3.3	Wie man die Arbeit an den Aufgaben organisiert	395
20.3.4	Welche Arbeit muss erledigt werden?	395
20.3.5	Wer erledigt die Arbeit?	396
20.4	Daily Scrum	396
20.5	Die Durchführung der Aufgaben – Technische Praktiken	397
20.6	Kommunizieren	398
20.6.1	Task Board	398
20.6.2	Das Sprint-Burndown-Chart	399
20.6.3	Das Sprint-Burnup-Chart	402
20.7	Abschließende Bemerkungen	403
21	Sprint Review	405
21.1	Überblick	405
21.2	Teilnehmer	406
21.3	Vorarbeiten	407
21.3.1	Entscheiden, wen man einlädt	408
21.3.2	Die Aktivität zeitlich planen	408
21.3.3	Bestätigen, dass die Sprint-Arbeit erledigt ist	409
21.3.4	Auf die Demonstration vorbereiten	410
21.3.5	Festlegen, wer was macht	410
21.4	Das Vorgehen	410
21.4.1	Zusammenfassen	411
21.4.2	Demonstrieren	412
21.4.3	Diskutieren	413
21.4.4	Ändern	413
21.5	Sprint-Review-Probleme	414
21.5.1	Abnahmen der PBIs	414

21.5.2	Sporadische Teilnahme	414
21.5.3	Umfangreiche Entwicklungsprojekte	415
21.6	Abschließende Bemerkungen	415
22	Die Sprint-Retrospektive	417
22.1	Überblick	417
22.2	Teilnehmer	419
22.3	Die Vorarbeit	420
22.3.1	Den Fokus der Retrospektive definieren	420
22.3.2	Die Übungen auswählen	421
22.3.3	Objektive Daten sammeln	421
22.3.4	Die Retrospektive strukturieren	422
22.4	Das Vorgehen	423
22.4.1	Die Atmosphäre gestalten	424
22.4.2	Gemeinsamer Kontext	425
22.4.3	Einsichten identifizieren	427
22.4.4	Aktionen festlegen	430
22.4.5	Die Retrospektive schließen	433
22.5	Dranbleiben	433
22.6	Probleme der Sprint-Retrospektive	434
22.7	Abschließende Bemerkungen	436
23	Der Weg nach vorn	437
23.1	Es gibt keinen Endzustand	437
23.2	Finden Sie Ihren eigenen Weg	438
23.3	Best Practices mit anderen teilen	438
23.4	Mit Scrum den Weg nach vorn finden	439
23.5	Immer weiter!	441
A	Referenzen	443
B	Glossar	447
	Stichwortverzeichnis	469