

5. Auflage



boris GLOGER

Scrum

PRODUKTE ZUVERLÄSSIG
UND SCHNELL ENTWICKELN

HANSER

Bleiben 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



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Boris Gloger

Scrum

Produkte zuverlässig und schnell
entwickeln

5., überarbeitete Auflage

HANSER

Boris Gloger, Geschäftsführer borisgloger consulting GmbH und borisgloger professionals gmbh,
Wien – Baden-Baden
Kontakt: boris.gloger@borisgloger.com

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 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 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.

© 2016 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Brigitte Bauer-Schiewek

Copy editing: Dolores Omann, Wien

Herstellung: Irene Weilhart

Layout: Manuela Treindl, Fürth

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

Umschlagrealisation: Stephan Rönigk

Druck und Bindung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-44723-3

E-Book-ISBN: 978-3-446-44836-0

Inhalt

Vorwort zur 1. Auflage..... XI

Vorwort zur 5. Auflage..... XIII

Über den Autor..... XV

1 Prinzipien, Geschichte(n), Hintergründe 1

1.1 Was ist Scrum? 5

1.1.1 Eine kurze Einführung in die Funktionsweise 8

1.1.2 Rollen, Meetings, Artefakte 11

1.2 Ein Begriff - viele Einsatzmöglichkeiten 14

1.3 Scrum - eine Bewegung entsteht 19

1.4 Warum Scrum funktioniert 25

1.4.1 Ford und Sloan - die Ursprünge der Massenfertigung 27

1.4.2 Lean Manufacturing 28

1.4.3 Lean Product Development 31

1.4.4 Second Generation Lean Product Development 32

1.5 Durch teamzentriertes Arbeiten zur modernen Wissensorganisation 36

1.5.1 Kontrollierbarkeit des Unkontrollierbaren 45

1.5.1.1 Systemische Grundlagen 47

1.5.1.2 Das Scrum-Team und Selbstorganisation 49

1.5.2 Kontinuierliche Verbesserung - Fast Feedback 51

1.5.3 Höhere Produktivität 54

1.5.4 Freude an der Arbeit macht leistungsfähig 55

2 Die Rollen - klare Verantwortlichkeiten 59

2.1 Das Entwicklungsteam - die Spezialisten 64

2.1.1 Wie baut man ein Scrum-Team? 70

2.1.2 Die Phasen der Teambildung 74

2.1.3 Probleme des Teams bei der Implementierung 75

2.2 Der Product Owner 77

2.2.1 Das Product Backlog 79

2.2.2 Das Produkt annehmen, verbessern oder ablehnen 81

2.2.3 Den Releaseplan bestimmen und managen 81

2.2.4	Die Verbindung zwischen Product Owner und Entwicklungsteam.....	82
2.2.5	Den Return on Investment bestimmen und sichern.....	82
2.2.6	Wer sollte die Rolle des Product Owners übernehmen?.....	83
2.2.7	Der Product Owner im skalierten Umfeld.....	83
2.3	Der ScrumMaster – ein Change Agent.....	85
2.3.1	Scrum implementieren.....	86
2.3.2	Das Abarbeiten von Impediments.....	87
2.3.3	Die Arbeit mit dem Entwicklungsteam.....	88
2.3.4	Die Arbeit mit dem Product Owner.....	92
2.3.5	Die Steigerung der Produktivität.....	93
2.3.6	Hat der ScrumMaster einen Fulltime-Job?.....	94
2.3.7	Wer wird ScrumMaster?.....	97
2.4	Der Customer – der Finanzier.....	98
2.5	Der User.....	100
2.6	Das Management – die Stabilisatoren der Organisation.....	101
2.7	Die Rollen ausüben und klar trennen.....	105
3	Strategisches Planen in Scrum.....	109
3.1	Was ist Planen?.....	111
3.2	Planungsebenen: Strategie und Taktik.....	115
3.3	Die Vision.....	116
3.3.1	Der Product Owner formuliert die Vision.....	116
3.3.2	Wie erschafft man eine Vision?.....	117
3.4	Führungsaufgabe Visionsgenerierung.....	120
3.5	Constraints festlegen.....	121
3.6	Die User-Rolle ist tot – es lebe die Persona!.....	122
3.7	Das Product Backlog.....	124
3.7.1	Hilfsmittel für das Verwalten des Product Backlogs.....	125
3.7.2	Product Backlog für große Teams und Multiteams.....	127
3.7.3	Was ist ein Product Backlog Item?.....	128
3.7.4	Product Backlog Items als User Storys formulieren.....	130
3.8	Priorisierung des Backlogs.....	131
3.8.1	Grundlage der Priorisierung: Business Value.....	132
3.8.2	Methoden der Priorisierung.....	133
3.9	Schätzen in Scrum.....	138
3.9.1	Vorhersagbarkeit und Schätzungen.....	139
3.9.2	Schätzen mit Storypoints.....	141
3.9.3	Magic Estimation.....	143
3.9.4	Die Velocity bestimmen.....	145
3.9.5	Der Releaseplan.....	146
3.10	Die Planung geht weiter.....	147
3.11	Die Zusammenhänge zwischen strategischer und taktischer Planung.....	148
4	Der Sprint – das Produkt entsteht.....	151
4.1	Die grundlegenden Prinzipien des Sprints.....	153
4.2	Das Estimation Meeting.....	154

4.2.1	Durchführung des Estimation Meetings	155
4.2.2	Das Estimation Meeting mithilfe von Magic Estimation	156
4.3	Das Sprint Planning – taktisches Planen	158
4.3.1	Sprint Planning Meeting 1 – Briefing und Analyse	160
4.3.2	Sprint Planning Meeting 2 – Design	163
4.3.3	Sprint Planning mit großen oder mehreren Teams – Skalierung	167
4.4	Das Daily Scrum – tägliche Synchronisation (reloaded)	171
4.4.1	Das neue Daily Scrum – Version mit Taskboard	172
4.4.2	Das neue Daily Scrum – Version ohne Taskboard	173
4.4.3	Mögliche Probleme im Daily Scrum	174
4.4.4	Daily Scrum für große und verteilte Teams	175
4.5	Sprint Review – das Produkt vorstellen	177
4.5.1	Die Bedeutung von „fertig“	177
4.5.2	Ablauf und Regeln des Sprint Reviews	179
4.5.3	Konsequenzen aus dem Sprint Review	180
4.5.4	Das Sprint Review im skalierten Umfeld	181
4.6	Sprint-Retrospektive – kontinuierliches Verbessern	182
4.6.1	Warum funktionieren Retrospektiven?	183
4.6.2	Lernen: ent-täuschte Erwartungen	184
4.6.3	Die sechs Schritte der erfolgreichen Retrospektive	185
4.7	Der Sprint selbst – zwischen den Meetings	194
4.7.1	Der Ablauf des Sprints – Kommunikation, Kommunikation, Kommunikation	194
4.7.2	Gemeinsamer Fokus	195
4.7.3	Die Aufgabe des Teams im Sprint	195
4.7.4	Die Aufgabe des Product Owners im Sprint	196
4.7.5	Die Aufgabe des ScrumMasters im Sprint	197
4.7.6	Was kann während eines Sprints passieren?	197
4.7.7	Wann kann ein Sprint abgebrochen werden?	198
4.7.8	Konflikte – es menschelt	199
4.7.9	Verlängerung des Sprints	201
4.7.10	Die Scrum Engine – zermürbende Monotonie	202
5	Reporting – wissen, wo wir stehen	205
5.1	Das Sprint Burndown-Chart	207
5.2	Das Taskboard	209
5.3	Das Story Burndown-Chart	209
5.4	Das Release Burndown-Chart	211
5.5	Das Parking-Lot-Chart	212
5.6	Das Velocity-Chart	213
5.7	Das Logbuch	214
5.8	Das Impediment Backlog	215
5.9	Die Retrospektive	215
5.10	Das Sprint Review	216
5.11	Berichten im skalierten Umfeld	216
5.12	Elektronische Hilfsmittel	218

6	Professionalität: Test, Integration, Release	219
6.1	Professionalität und Risiko	219
6.2	Auswirkung der schlechten Qualität	225
6.3	Entwicklungspraktiken der Balanced Agility	227
6.3.1	Kontinuierliche Integration – das Produkt entsteht	227
6.3.2	Qualität – testen, testen, testen	228
6.3.3	Release-Durchführung – das Produkt zur Verfügung stellen	230
6.4	Kontinuierliches Liefern	232
7	Scrum skalieren	233
7.1	Die Prinzipien skaliert	234
7.2	Skalieren ohne Blueprints	238
7.2.1	Das skalierte Daily Scrum und das Scrum of Scrums	240
7.2.1.1	Das teamübergreifende Scrum-Board	243
7.2.1.2	Der Wert des Scrum of Scrums	245
7.2.1.3	Das ScrumMaster Daily	247
7.2.1.4	Das Product Owner Daily	248
7.2.2	Wöchentliche Meetings	249
7.2.2.1	Das ScrumMaster Weekly	250
7.2.2.2	Das Product Owner Weekly – Business Value Estimation Meeting	251
7.2.3	Skaliertes Estimation Meeting	252
7.3	Die Rollen skaliert	253
7.3.1	Das ScrumMaster-Team	253
7.3.2	Das Product-Owner-Team	255
7.3.3	Die Gilden – Communities of Practice	256
7.3.4	Support-Teams	259
7.4	Skalieren über die Architektur	263
7.5	Verteilte Standorte	265
7.5.1	Das verteilte Team ist zu groß	266
7.5.2	Zu viele Meetings	267
7.5.3	Elektronische Tools	268
7.5.4	Das Linienmanagement	269
7.5.5	Visionen und Rahmenbedingungen	270
7.5.6	Wo sitzt der Product Owner?	272
7.5.7	Wo sitzt der ScrumMaster?	273
7.5.8	Freiberufler oder Mitarbeiter anderer Unternehmen im Team	276
8	Leadership, Emotion, Kreativität	281
8.1	Leadership	281
8.1.1	Arbeit an der eigenen Persönlichkeit	285
8.1.2	Circle of Safety	286
8.1.3	Entscheidungen einfordern	288
8.1.4	Fokus auf das Funktionierende	289
8.1.5	Positives bemerken	291
8.2	Emotionen	292
8.3	Kreativität	293

9 Scrum – Management mit Werten	299
9.1 Fokus auf das, was wichtig ist.	301
9.2 Commitment – einander vertrauen.	302
9.3 Offenheit – aufeinander zugehen	303
9.4 Respekt – Unterschiede würdigen	305
9.5 Mut – neue Wege einschlagen.	308
9.6 Das Produkt ist mehr als Strategie und Taktik	309
9.7 Scrum – eine wertvolle Erfolgsgeschichte	310
10 Fallstudien.	313
10.1 Holtzbrinck Legal/General Counsel: die agilen Anwälte	313
10.1.1 Mit Kanban beim Status quo starten	314
10.1.2 Daily Scrum und Retrospektive	315
10.1.3 Die Rechtsabteilung wird zur Serviceeinheit	316
10.1.4 Wissenstransfer	317
10.2 EWE: Energiekonzern mit agilem Antrieb	319
Literaturverzeichnis.	321
Stichwortverzeichnis.	325

Vorwort zur 1. Auflage

Warum nutze ich Scrum, bin Scrum-Trainer, habe eine Consulting-Firma, die Kunden hilft, Scrum zu nutzen, und wieso versuche ich selbst, die Prinzipien, die hinter Scrum stecken, zu verwirklichen? Vielleicht ist es der gleiche Grund, der Sie bewegt, dieses Buch in die Hand zu nehmen: Ich bin unzufrieden.

Ich sehe jeden Tag, wie intelligente und gut ausgebildete Menschen zur Arbeit gehen und an ihrer Arbeit keine Freude mehr empfinden. Menschen wechseln den Job und hoffen, dass dieser Job die erhoffte Befriedigung bringt. Und am Ende sind sie wieder gelangweilt, schalten ab und wollen am liebsten zu Hause bleiben.

Ich habe gesehen, dass Software-Entwicklungsprojekte scheiterten, weil sich niemand verantwortlich gefühlt hatte. Es ist absurd: Projekte, an denen Dutzende Menschen hart und lange gearbeitet hatten, scheiterten, weil am Ende alle allein die Schuld gegeben hatten und nur noch jeder daran interessiert war, seinen eigenen Bereich zu verteidigen. Obwohl jeder Beteiligte früh sah, dass sich das Projekt auf dem Holzweg befand, wurde es fortgeführt, man ging mit offenen Augen auf den Abgrund zu.

Ich sehe Menschen, die ihr Leben im privaten Alltag meistern. Aber wenn sie morgens in ihre Firma gehen, werden sie dort zu ängstlichen Wesen. Menschen, die genau wissen, was sie können, trauen sich nicht, ihren Chefs zu sagen, dass die Anweisungen, die sie erhalten, sinnlos und unproduktiv sind. Sie bekommen in Besprechungen Schweißausbrüche, wenn sie etwas sagen sollen, oder sie sind froh, wenn sie den ganzen Tag nicht von ihrem Chef wahrgenommen werden.

Ich sehe Chefs von Software-Entwicklungsabteilungen resignieren, weil ihre Mitarbeiter nicht motiviert sind, nicht tun, was getan werden müsste, und abends lieber fluchtartig nach Hause gehen, als fünf Minuten länger in der Firma zu sitzen. Das Resultat sind Chefs, die in ihrer Verzweiflung beginnen, Kontrollmechanismen aufzubauen, die die begonnene Abwärtsspirale verstärken.

Das machte mich unzufrieden, und ich wollte das ändern. Ich dachte am Anfang, es läge an der falschen Art und Weise, Software-Entwicklung zu betreiben, an den falschen Methoden und Tools. Ich dachte, es läge an den falschen Managementmethoden oder daran, dass an den entscheidenden Stellen die falschen Leute sitzen. In der Rolle des Business-Analysten eines Entwicklungsteams vermutete ich zuerst, dass es am unterschiedlichen Sprachgebrauch von Software-Entwicklern und Fachabteilungen liegen musste und dass es doch einen Weg geben müsste, das babylonische Sprachgewirr aufzulösen. Kurz: Ich war auf der Suche nach der Methode, mit der man es richtig machen kann.

Dann, Jahre später, fand ich die Methode: Scrum. Auf den ersten Blick schien es so, als wäre Scrum die Antwort auf all die Fragen, die ich so hatte. Schnell stellte sich heraus, dass Scrum

diese Antworten nicht liefern konnte. Scrum zeigte mir aber einen Weg, wie man die Antworten finden konnte, und machte klar, wohin man schauen musste, wenn man Probleme in der Software-Entwicklung lösen wollte. Scrum zeigte nicht, wie ich es richtig machen sollte. Scrum zeigte mir nur ständig, wo ich noch etwas tun musste. Ich musste meine Management-Skills verbessern, Mitarbeiter kündigen, neue Entwicklungsmethoden einführen, mich mit meinen Chefs anlegen – einmal zog ich die Konsequenzen selbst. Also alles in allem Dinge, die unbequem sind, die man nicht unbedingt gerne tut und die viel harte Arbeit erfordern. Scrum machte es mir und vielen meiner Mitarbeiter also nicht leichter, sondern es schien, als würde es sogar schwerer werden. Aber uns fiel auf, dass sich diese Schwierigkeiten mit dem Erlernen und Trainieren einer neuen Sportart vergleichen ließ. Es machte Spaß und war befriedigend. Zwar ging es nicht ohne Aufwand und Mühe, ohne hartes Training, ständiges Wiederholen von bestimmten Übungen und ständiges Überprüfen und Evaluieren unseres Fortschrittes. Aber wir wurden belohnt. Unsere Produktivität stieg, wir hatten mehr Freude an der Arbeit, und unsere Chefs wurden mit jeder Woche zufriedener mit uns.

In diesem Buch möchte ich Ihnen zeigen, wie auch Sie diese Erfolge erzielen können. Scrum wird Ihnen einfach erscheinen und doch werden Sie ständig die Frage im Kopf haben: „Und wie funktioniert das bei mir?“ Sie werden sich fragen, ob Scrum für Sie anwendbar ist und ob es funktioniert.

Ich kann Ihnen diese Frage beantworten. Scrum und seine Praktiken sind erfolgreich. Hunderte von Firmen und Tausende von Projektteams nutzen Scrum und sind im Entwickeln von Software erfolgreicher geworden. Haben diese Teams noch Probleme? Ja, viele! Sie finden sogar ständig neue Probleme. Aber sie haben sich auf den Weg gemacht. Daher wissen sie, wie viel noch zu tun ist. Denn erst wenn man auf dem Weg ist, bemerkt man, wie viel Wegstrecke vor einem liegt. Wenn Sie für einen Marathon trainieren, bemerken Sie auch erst nach dem Beginn des Trainings, wie hart der Weg wirklich ist.

Dieses Buch enthält die Ideen von unzähligen Personen, denen ich zutiefst dankbar bin. An erster Stelle Ken Schwaber und Jeff Sutherland, die uns allen gezeigt haben, wie einfach es ist, das Chaos zu kontrollieren. Norman Kerth, der mir das Wesen der Projekt-Retrospektive gezeigt hat. Jean Tabaka, Stacia Broderick, Hubert Smith, Jens Østergard und vielen anderen aus der Scrum Community, die einerseits den Weg mit mir gemeinsam gegangen sind und mir andererseits durch viele Gespräche geholfen haben, Scrum besser zu verstehen.

Ich möchte meinem ersten Scrum-Team danken: Gillian, Sven, Graig, Jakob, Niki und Marc sind mit mir damals auf die Reise gegangen. Und ich danke meinem damaligen Chef Michael Schindelar, der uns einfach ausprobieren ließ, wie Scrum funktioniert.

Mein besonderer Dank gilt meinen Freunden und Mitarbeitern, die heute mit mir Scrum in der Form leben, wie ich es verstehe und wie es in diesem Buch beschrieben ist. Nicht zu vergessen die Teilnehmer der Trainings aus vier Jahren, die alle ihren Beitrag dazu geleistet haben, dass dieses Buch, so wie es vor Ihnen liegt, geschrieben werden konnte.

Scrum ist eine Reise mit dem Ziel, seine Fähigkeiten zu erkennen, Erfahrungen zu sammeln, dabei Teamgeist entstehen zu lassen und zu erfahren, Höhen und Tiefen zu erleben, um am Ende ein Produkt zu liefern, das uns selbst und unsere Kunden begeistert, auf das wir stolz sind, weil wir uns einbringen konnten, und das als Ergebnis ein Teil von uns ist.

Viel Spaß beim Lesen!

Vorwort zur 5. Auflage

Acht Jahre sind vergangen, seit ich die ersten Zeilen für dieses Buch geschrieben habe. Acht Jahre Scrum, agile Softwareentwicklung, agiles Projektmanagement, agile Organisationsentwicklung – die Welt hat sich für viele Unternehmen verändert, denn die Ideen von Jeff Sutherland und Ken Schwaber sind durchgedrungen. Heute diskutieren wir mit unseren Kunden nicht mehr darüber, ob Scrum erfolgreich ist oder überhaupt funktionieren kann, sondern nur noch über das Wie im jeweiligen Kontext. Mit Hilfe von Scrum wird heute in den Vorstandsetagen über Unternehmensstrategien nachgedacht, es wird als Methode des Change Managements genutzt und selbst Hardware wird heute von Scrum-Teams entwickelt. Ikujiro Nonaka selbst antwortete 2012 bei einem Vortrag an der Wirtschaftsuniversität Wien auf die Frage, wie Unternehmen in Zukunft agieren sollten: „Macht Scrum!“ Und mittlerweile hat sich im Topmanagement herumgesprochen, dass Scrum dabei helfen kann, neue Antworten auf die immer gleichen Fragen zu finden: Wie wird das Unternehmen in Zeiten des globalen Wettbewerbs noch leistungsfähiger? Wie lassen sich die Mitarbeiter halten? Wie wird das Unternehmen im War for Talents zu einem attraktiven Arbeitgeber?

Acht Jahre nach der ersten Ausgabe dieses Buchs lag mir die mittlerweile fünfte Überarbeitung wie ein Stein im Magen. Sollte ich wieder nur ein paar Sätze austauschen, da und dort etwas deutlicher werden oder zumindest an einigen Stellen großzügig neue Erkenntnisse einfließen lassen? Ich habe mich dazu entschieden, ganze Passagen neu zu schreiben, überflüssige Metaphern rauszuwerfen und in manchen Kapiteln massiv einzugreifen – etwa in Kapitel 7 zur Skalierung oder Kapitel 8 zur Führung. Vielleicht empfinden Sie meine Grafiken als Kritzeleien. Aber mir geht es um die Veranschaulichung, nie um das schöne Bild.

Alles in allem ist es mir wichtig, Ihnen ein modernes Scrum zu präsentieren: Mit neuen Ideen, weniger Pathos als bei der ersten Auflage, als oft meine eigene Begeisterung für das Thema mit mir durchgegangen ist, aber dafür mit noch mehr Beispielen, gelebten Ideen und Beiträgen meiner Kollegen Christoph Schmiedinger und Anton Jessner. Die beiden arbeiten jeden Tag mit Teams und räumen für sie Impediments weg, engagieren sich für sie beim Management und zeigen ihnen, wie das neue Arbeiten mit Scrum gelingt.

Dieses Buch lebt in der fünften Auflage von der Leistung meines Teams bei borisgloger consulting. Meine Kolleginnen und Kollegen zeigen Tag für Tag, dass unsere Ideen funktionieren, und ihre Kritik ist mir wichtig. Anders als bei der ersten Auflage bin ich heute in meinem eigenen Unternehmen von Experten umgeben, die beurteilen können, ob das, was zwischen den Buchdeckeln steht, gelebte Praxis ist oder beliebige Fantasien sind. Sie haben es für gut befunden und nutzen dieses Buch bei der Ausbildung unserer neuen Kollegen. Es steht also wirklich das drin, was uns in unserer Arbeit weiterbringt. Danke euch allen für euer Engagement!

Ein dickes Dankeschön geht an Dolores Omann, die wie bei jedem meiner Bücher schonungslos offenlegt, wenn ich nicht mehr einfach und klar schreibe, die jeden Satz fünfmal umdreht und dafür sorgt, dass meine Ideen lesbar werden.

Ein Dankeschön an meine persönliche Assistentin Janine Frensemeyer. Sie hat mir Zeit freigeschaufelt, damit ich mich wirklich ans Umschreiben machen konnte. Ich wäre nie weitergekommen, wenn sie nicht immer wieder gesagt hätte: „Nein, da machen wir keinen Termin, da wolltest du schreiben!“

Vielen Dank an Christoph Schmiedinger und Anton Jessner, die beide ein Kapitel für diese Auflage beigesteuert haben. Herzlichen Dank an EWE und Holtzbrinck für die Erlaubnis, die in diesem Buch enthaltenen Fallstudien veröffentlichen zu dürfen. Das macht ein solches Buch extrem wertvoll für die Praxis, denn es zeigt: Die neue Welt des Arbeitens existiert.

Schließlich gilt mein Dank meiner Frau Kathrin und meiner Familie. Vielen Dank, dass ihr mich so unterstützt.

Aber diese 5. Auflage könnte gar nicht erscheinen, wenn es nicht Sie, die Leser meiner Bücher und unseres Blogs blog.borisgloger.com gäbe. Danke, dass Sie die Ideen verbreiten und vor allem selbst umsetzen! Hätten Sie, hättet ihr nicht an die Ideen aus diesem Buch geglaubt und hättet ihr sie nicht ausprobiert, wäre Scrum in Deutschland, Österreich und der Schweiz nicht zu dem geworden, was es heute ist: Ein Weg für gelingende Wissensarbeit im 21. Jahrhundert. Macht weiter so und habt viel Spaß beim Lesen!

Laxenburg im April 2016

Boris Gloger



Die Scrum-Checkliste

Für die Scrum-Checkliste besuchen Sie bitte

www.borisgloger.com

Dort gibt es die Checkliste auf Deutsch beziehungsweise Englisch und alle Informationen über die Checklist-App.

Über den Autor

Boris Gloger führte 2002 sein erstes Scrum-Team beim österreichischen Mobilfunker ONE zum Erfolg. Als weltweit erster, von Ken Schwaber ausgebildeter Certified Scrum Trainer hat er wesentlich dazu beigetragen, dass sich Scrum in Europa, Südafrika und Brasilien als Standard der agilen Softwareentwicklung durchgesetzt hat. Bevor er 2008 die Managementberatung borisgloger consulting GmbH mit Sitz in Baden-Baden gründete, war der Unternehmer als Business Analyst, Teamleader, Projektmanager und Scrum Consultant für globale Unternehmen (z. B. EDS, Nokia, BenQ) tätig. borisgloger consulting bietet Training und Consulting zur agilen Produkt- und Organisationsentwicklung mit Scrum sowie zum agilen Management für Fach- und Führungskräfte an. Boris Gloger gilt inzwischen als einer der progressivsten Denker im Bereich Management und Organisation und ist ein gefragter Vortragender auf Managementkonferenzen.



Folgende Bücher von Boris Gloger sind im Hanser Verlag erschienen:

- Selbstorganisation braucht Führung. Die einfachen Geheimnisse agilen Managements. Hanser, 2014.
- Wie schätzt man in agilen Projekten – oder wieso Scrum-Projekte erfolgreicher sind. Hanser, 2014.
- Der agile Festpreis. Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge. 2. Aufl. Hanser, 2014.
- Erfolgreich mit Scrum: Einflussfaktor Personalmanagement. Finden und Binden von Mitarbeitern in agilen Unternehmen. Hanser, 2011.

Kontakt: boris.gloger@borisgloger.com

1

Prinzipien, Geschichte(n), Hintergründe

*Haben Sie den Mut, Nein zu sagen.
Haben Sie den Mut, der Wahrheit ins Gesicht zu blicken.
Tun Sie das Richtige, weil es richtig ist.
Das sind die magischen Worte, mit denen Sie Ihr Leben mit Rechtschaffenheit leben.*

W. Clement Stone

In „Snow Crash“ entwirft Neal Stephenson das Bild einer Zukunft, in der die US-Regierung zum weltgrößten und gleichzeitig ineffizientesten Produzenten von Software geworden ist. Die besten Software-Engineering-Techniken werden eingesetzt. Diese sind grausam, aber extrem ausgeklügelt: Tausende Programmierer wissen zwar nicht mehr, warum sie ein Stück Software schreiben, aber sie entwickeln unglaublich viel davon. Architekten und Projektmanager entwerfen die Komponenten, zerteilen sie in immer kleinere Stücke und geben sie dann mit überbordender Dokumentation an die Programmierer, die nur diesen einen Job zu erledigen haben. Oft genug wird die Arbeit von Wochen durch neue Anweisungen zunichtegemacht. Mehr als die Hälfte ihres Arbeitstages verbringen die Programmierer damit, zu lesen, was von ihnen erwartet wird. Alles, was sie tun, wird exakt vermessen und beobachtet. Aus Hackern sind Fließbandarbeiter geworden [Stephenson 1992]. Von den Übertreibungen abgesehen hat Stephenson bereits 1992 eine Entwicklung vorausgesehen, die in vielen Unternehmen Realität geworden ist: Softwareentwicklung soll nach definierten Prozessen ablaufen und nach ingenieurwissenschaftlichen Methoden funktionieren.¹

Aber manchmal kommt es anders. Die Veränderung der Arbeit hat bereits begonnen. Erfolgreiche Unternehmen wie Google, Gore, 3M, Semco und Toyota machen vor, wie man das Potenzial seiner Mitarbeiter nutzt und sie nicht nur als Ressourcen sieht. Diese Unternehmen sind erfolgreich, weil sie Wege gefunden haben, um die Innovationskraft ihrer Mitarbeiter zu wecken, zu erhalten und zu nutzen. Sie setzen ebenso auf das Individuum wie auf das Team, dezentralisieren Entscheidungen und schaffen es, dass die Menschen gerne für sie arbeiten. Sie erreichen nicht nur den Verstand der Mitarbeiter, sondern auch ihr Herz und ihre Leidenschaft.² Führungskräfte auf der ganzen Welt suchen nach einer Methode, wie sie den Anforderungen der Globalisierung und der Beschleunigung begegnen können. Dabei reicht es nicht, einfach nur zu kopieren, was Google vormacht. Jedes Unternehmen, jede Führungskraft muss ihren eigenen

¹ <http://www.stevemccconnell.com/psd/04-senotcs.htm> (29.12.2015)

² Bei [Hamel 2007] findet sich dazu eine interessante Aufschlüsselung: „If we were to measure the relative contribution that each of these human capabilities [Anm.: passion, creativity, initiative, intellect, diligence, obedience] makes to value creation, recognizing we now live in a world where efficiency and discipline are table stakes, the scale would look something like this: Passion 35%, Creativity 25%, Initiative 20%, Intellect 15%, Diligence 5%, Obedience 0%.“

Weg finden. Es gibt jedoch ein Bündel von Regeln, das weltweit, in allen Kulturkreisen dazu führt, dass Unternehmen ihre Antwort auf die Anforderungen der heutigen Zeit finden: **Scrum**. Scrum ist vor allem ein Change-Management-Ansatz und ein Weg für das Team-, Abteilungs- und Organisationsmanagement. Die Regeln und Elemente von Scrum kann man nutzen, um Projekte zu steuern und Abteilungen und Unternehmen zum Erfolg zu führen. Das macht aus Scrum aber keine Methode und kein Prozessmodell. Scrum ist eine Einstellung zum Umgang mit Menschen und Mitarbeitern, Kunden und Managern. Scrum ist eine innere Haltung, die sich durch Disziplin und Verantwortungsbewusstsein auszeichnet.

Jeff Sutherland und Ken Schwaber sind einst angetreten, um eine neue Professionalisierung in der Softwareentwicklung zu beginnen. Sie wollten Entwicklungsmannschaften weltweit dazu befähigen, innerhalb weniger Wochen Software zu liefern. Mit ihrem Managementframework Scrum brachen sie mit dem Paradigma, dass man Projekte nur durch definierte Prozesse managen könne. Basierend auf vielen Ideen, die ich in diesem Buch vorstellen will, schufen sie ein empirisches Prozessmodell. In dessen Zentrum setzten sie eine Managementrolle, den **ScrumMaster**. Er (oder sie) hat die Aufgabe, Scrum zu implementieren, das Scrum-Team zur Produktivität zu befähigen und Probleme jeder Art aus dem Weg zu räumen. Damit wird der ScrumMaster zum Organisationsentwickler. Demgegenüber steht der **Manager**, der mit dem ScrumMaster gemeinsam die Organisation so umgestalten muss, dass Produkte schneller erzeugt werden können. Eine Voraussetzung muss der Manager dazu schaffen: Gemeinsam mit dem ScrumMaster muss er Rollen kreieren, die legitimiert sind, ihre Aufgaben möglichst autark, also eigenverantwortlich und selbstbewusst zu erfüllen. Bei diesen Rollen handelt es sich um das **Entwicklungsteam** und um den **Product Owner**, der den Produktentwicklungsprozess anführt.

Der ScrumMaster – ein machtloser Organisationsentwickler

Bevor wir tiefer darin eintauchen, was Scrum ist und wie es funktioniert, möchte ich die Rolle des ScrumMasters besonders beleuchten. Er hat im wahrsten Sinne des Wortes in all diesen Zusammenhängen eine Schlüsselrolle. Durch die Art und Weise, wie er seine Aufgabe versteht und lebt, kann er mehr bewirken, als ihm am Anfang vielleicht bewusst ist. ScrumMaster sind per definitionem Change Agents. Ihre Aufgabe ist es, den Status quo zu ändern. Wer den Status quo ändern will, steht außerhalb des Gesetzes. In der westlichen Geschichte gibt es einige berühmte Veränderer. Sokrates zum Beispiel war mit seinen Ideen eine Gefahr für die herrschende Kaste Athens. Wegen dieser Ideen musste er den Schierlingsbecher leeren. Er tat es freiwillig, weil er das Recht des Staates zur Bestrafung anerkannte. Nelson Mandela wirkte 24 Jahre lang aus dem Gefängnis für seine Ideen. Dabei rief er nie zum bewaffneten Widerstand auf. Gandhi war etliche Male im Gefängnis, wurde verfolgt und bestraft und nutzte die Regeln und die Moralvorstellungen der Engländer gegen sie selbst. Martin Luther King wurde wegen seines Kampfes für die Rechte der Schwarzen am 4. April 1968, fünf Jahre nach seiner berühmten Rede „I have a dream“, erschossen.

Sie alle hatten den inneren Drang, gegebene Verhältnisse zu ändern. Sie zeichneten sich durch die Fähigkeit, andere Menschen von ihren Ideen zu überzeugen, als Führungspersönlichkeiten und Veränderer aus. Sie erreichten ihre Ziele, indem sie Menschen dazu brachten, mit ihnen gemeinsam diesen Weg zu gehen, obwohl keine Autorität sie dazu eingesetzt hatte. Sie waren nicht von Organisationen oder Chefs darum gebeten worden, diese Ideen durchzusetzen. Norman Kerth bezeichnet diese Art der Führung in seinem Buch „Project Retrospectives“ als „Leading from the Position of No-Power“ [Kerth 2001]. Diese Persönlichkeiten bezogen die

Rechtfertigung für ihr Handeln aus den Gesetzen, an die ihre Gegner selbst glaubten. Daher wurde ihnen von den Menschen, die ihnen folgten, die Macht gegeben, sie zu führen. Am Beispiel von Martin Luther King kann man das sehr gut sehen. Er verankerte seinen Traum im „American Dream“. In seiner berühmten Rede am 28. August 1963 in Washington sagte er³:

„I say to you today, my friends, so even though we face the difficulties of today and tomorrow, I still have a dream. It is a dream deeply rooted in the American dream. I have a dream that one day this nation will rise up and live out the true meaning of its creed: ‚We hold these truths to be self-evident: that all men are created equal.‘“

„I have a dream“ – mit diesen Worten drückte ein Mann seine Überzeugung in einer Form aus, die Millionen von Menschen berührte und bewegte. Nicht weil er einen Traum hatte, der mit dem Traum der übrigen, weißen Amerikaner nichts zu tun hatte. Nein, weil er die Emotionen ansprach, die jeder Amerikaner mit diesem Traum verbindet. Seine Überzeugung, verwurzelt im Traum aller US-Amerikaner, trug ihn und die Menschen, die an ihn glaubten, zu neuen Ufern und änderte den Status quo.

Natürlich vergleiche ich den ScrumMaster, der ein Team oder eine Abteilung anführt, nicht mit Martin Luther King oder Nelson Mandela. Ich will aber an dieser Stelle ein Bild zeichnen, das eindringlich zeigt, was ein ScrumMaster ist:

Er ist eine Führungspersönlichkeit, die verändern will. Er ist ein Change Agent, der die Macht, die Gegebenheiten zu ändern, nicht aus seiner Position bezieht, sondern aus seiner Überzeugung und aus dem Rückhalt, den er von den Menschen bekommt, für die er sich einsetzt.

Dazu muss er an die Richtigkeit seiner Handlungen glauben. Der ScrumMaster wird sich so oft gegen den Status quo seines Unternehmens auflehnen, dass er bald als Querulant und Unruhe-stifter gelten wird. Gleichzeitig kann er auf diese Weise Verbesserungen erreichen, die viele andere vor ihm nicht erreicht haben und vielleicht nie erreichen werden. Jeder ScrumMaster muss aber ständig damit rechnen, dass er die Konsequenzen seines Handelns zu tragen hat.



Das Ziel vor Augen

Ein Scrum-Team bekam vom Management die Anweisung, es solle testgetrieben mit einem bestimmten Tool entwickeln. Im Scrum-Team selbst hatte aber niemand den entsprechenden Zugang. Auch von den Kollegen bekam das Team diesen Zugang nicht. Daraufhin beschloss das Scrum-Team, ein anderes Tool einzusetzen. Die Entscheidung des ScrumMasters, diesen Weg an der Management-Direktive vorbei zu unterstützen, war völlig korrekt. Oberstes Ziel war, nach der Iteration getesteten und fehlerfreien Code bereitzustellen. Die Vorgabe, dafür ein Tool aus der Organisation zu nutzen, war zu diesem Zeitpunkt nicht zu erfüllen. Das Team eskalierte sein Problem und blieb bis zur Lösung des Problems nicht stehen, sondern implementierte parallel dazu die notwendigen Tests im alternativen Tool. Das erzeugte einerseits den nötigen Druck auf die Organisation, andererseits wurde auf diese Weise das Sprint-Ziel erreicht. Der ScrumMaster wählte diese „Schlacht“, weil er sie nur gewinnen konnte. Er demonstrierte die Performance und den Willen seines Teams, die Vorgabe zu erfüllen, testgetrieben zu implementieren, und zeigte, dass die Organisation nicht in der Lage war, das Team entsprechend zu unterstützen. Er hatte das Richtige gegen die Regeln der Organisation getan.

³ Siehe: www.usconstitution.net/dream.html (29.12.2015)

Eines muss der ScrumMaster also bedenken: Er steht immer mit einem Bein auf der Straße! Indem er den Status quo ändert, riskiert er ständig seinen Auftrag oder seinen Job. Ken Schwaber sagt seit 2003 in seinen Trainings:

„A dead ScrumMaster is a useless ScrumMaster!“

Er unterstreicht damit, dass ein ScrumMaster einerseits die Organisation ändern muss, wenn er sein Team produktiver machen will. Andererseits muss er aber seine Schlachten sehr bewusst und überlegt führen. Muss er dennoch gehen, dann beglückwünschen wir in der Scrum Community diesen ScrumMaster, der in Ausübung seiner Pflicht „fällt“. Doch es gibt auch Hoffnung: Sind ScrumMaster erfolgreich, werden sie von den Organisationen belohnt. ScrumMaster, die funktionierende Scrum-Teams kreieren, haben in der Regel gute Chancen, befördert zu werden.

Bessere Produktentwicklung. Akzeptieren wir die Vorstellung, dass ein ScrumMaster ein Change Agent ohne Machtbefugnis ist, wird klar, was Scrum auch ist: eine *Change-Management-Methode*. Scrum einzuführen, heißt immer auch, die Organisation im Hinblick auf eine produktivere Produktentwicklung zu ändern. Scrum ist das Managementframework, mit dem diese Veränderung vorangetrieben werden kann, denn es legt wie ein Detektor die in der Organisation verborgenen Hindernisse offen, die bis dato die Produktentwicklung verlangsamt haben – zum Beispiel unklare Anforderungen, Unwissen des Teams, ständige Änderungen während der Implementierung, langes Warten auf Testsysteme etc. Der ScrumMaster benutzt sein Hauptwerkzeug, das Impediment Backlog, um notwendige Änderungen systematisch einzubringen und zu verfolgen. Scrum als Change-Management-Methode zeigt den Weg zur Veränderung auf, gehen müssen ihn aber alle Beteiligten – ScrumMaster, Manager und viele andere – gemeinsam.

Scrum verändert die Organisation entlang ihres Produkts „on-the-fly“: Es liefert nicht die Antworten darauf, wie spezielle Elemente einer Organisation mit Scrum umgesetzt werden sollen, sondern hinterfragt, ob diese Elemente sinnvoll und produktiv sind. Aus dem Anspruch, nach jedem Sprint an die Organisation ein fertiges Stück Produkt zu liefern, resultiert das moralische Recht auf die Veränderung. Es werden alle Arbeitsprozesse eines Teams und eines Projekts hinterfragt und die Elemente des Prozesses identifiziert, die *nicht wertschöpfend* sind. Es gilt, diese abzuschaffen und den Arbeitsprozess zu verbessern.

Von vielen Seminarteilnehmern, denen ich erklärt habe, was Scrum wirklich ist, höre ich: „Aha, dann kann man Scrum nur machen, wenn die Organisation entsprechend aufgestellt ist.“ Dahinter stecken die Erkenntnis, dass die Organisation verändert wird, und ein Ohnmachtsgefühl. Das Gegenteil ist der Fall. Scrum stößt nebenbei die Veränderung an, während das Produkt geliefert wird. Scrum verfolgt nicht das Ziel, Organisationen zu verändern. Scrum verändert Organisationen durch das permanente Liefern von Produktteilen, durch einen anderen Umgang mit Menschen und durch das Einführen von Disziplin und klaren Rollenverteilungen.

Ein Mensch, der die Rolle des ScrumMasters wählt, weiß das und erkennt seine Rolle als Change Manager. Er erkennt, dass es sich bei den meisten Problemen, die er mit Scrum aufdeckt, um Probleme handelt, die bereits vorher bekannt waren, nun aber offensichtlich und transparent werden. Das gibt ihm die Chance, die Probleme offen anzugehen.

■ 1.1 Was ist Scrum?



Ein großer, hell erleuchteter Besprechungsraum. Niko tritt ein. Seine Gedanken kreisen um die neu eingeführte Spiegelreflexkamera. Das Flaggschiffprodukt hat in den letzten Monaten wichtige Marktanteile verloren, da der Hauptkonkurrent eine neue DSLR herausgebracht hat. Sein Managementteam hatte ihm, dem CEO, versichert, dass die Menschen, die er jetzt in diesem Raum antreffen würde, die besten Spezialisten des Unternehmens seien. Er betritt den Meetingraum, stellt sich in die Mitte, seine Assistentin startet den Beamer, im Hintergrund ist ein Bild der Konkurrenzkamera zu sehen.

„Guten Morgen, meine Damen und Herren.“ Niko schaut in die angespannten Gesichter der fünf Männer und drei Frauen. Sie kommen aus allen Fachbereichen: Softwareentwicklung, Produktion, Gehäusebau und Hardware, Linsenspezialisten sind dabei und natürlich ein Profifotograf. „Ihre Anspannung ist verständlich“, denkt Niko, denn die Anwesenden wissen nicht, wieso sie eingeladen wurden. Sie haben erst vor 30 Minuten die Aufforderung erhalten, sofort in diesen Besprechungsraum zu kommen.

„Sie fragen sich wahrscheinlich, warum Sie hier sind. Die Antwort ist: Wir brauchen genau Sie für unser neues Projekt.“ Er deutet auf die Präsentation: „Hier sehen Sie die neue Kamera unseres Mitbewerbers. Wir verlieren Marktanteile. Alle Prognosen zeigen, dass wir hohe Verluste einfahren werden, wenn wir den Profifotografen keine Alternative bieten.“

Niko macht eine Pause. Er beobachtet die Frauen und Männer. Ihre Augen sind weit geöffnet, einige beugen sich nach vorne, andere richten sich leicht auf. Niko schaut wieder in die Runde und sagt dann: „Sie alle wissen, wie man eine bessere, schnellere und attraktivere Kamera bauen kann, als wir sie hier sehen. Ihre Aufgabe ist es, eine neue, bessere Kamera als diese zu bauen. Innerhalb von neun Monaten, und sie darf im Laden nicht mehr kosten als das Produkt der Konkurrenz.“

Die Frauen und Männer sehen sich an. Niko fährt fort: „Draußen wartet ein Bus, der Sie in Ihr neues Büro und Ihre neuen Laboratorien bringen wird. Damit Sie niemand stören kann, haben wir entschieden, dass Sie Arbeitsplätze bekommen, an denen Sie gemeinsam arbeiten können.“

Wieder macht er eine Pause und setzt dann fort: „Wir zählen auf Sie und wir erwarten von Ihnen in neun Monaten ein Produkt. Kein Positionspapier, keine Studie und keinen Prototyp. Wir wollen in neun Monaten eine neue Kamera vorstellen, die einen Tag später an die Händler ausgeliefert wird.“

Die hier dargestellte Szene beschreibt die Ausgangslage von Scrum. Ein Team (wir nennen es das Scrum-Team) zieht aus, um eine Mission zu erfüllen. Ein Team aus Spezialisten, von einer Vision geleitet, arbeitet sich diszipliniert und hoch professionell von einem Teilprodukt zum nächsten. Woche für Woche werden Teilergebnisse an den Nutzer des Produkts geliefert, bis schließlich das gesamte Produkt zur Verfügung steht.

Scrum ist auf der Metaebene eine *Überzeugung*, eine *Philosophie*, und auf der Prozessebene eine *Arbeitsweise* mit klar definierten Rollen und einem sehr einfachen Prozessmodell mit einem klaren und einfachen Regelwerk.

Sie werden in diesem ersten Kapitel aber verstehen lernen, warum Scrum weit mehr als das Prozessmodell ist. Sie werden erfahren, dass Scrum ein *Managementframework* für das Erstellen von Produkten ist, das – weit über die Produktentwicklungsteams hinaus – die Struktur und die Arbeitsweise einer ganzen Organisation beeinflusst. Ich werde Ihnen auch schildern, dass Scrum die Architektur der Produkte beeinflusst, dass Scrum dafür sorgt, dass sich Teams entwickeln und dass Scrum dabei hilft, in Organisationen ein neues Managementverständnis zu etablieren.

Woher stammt der Begriff?

Der Begriff Scrum ist dem Rugby entlehnt. Auf Deutsch bedeutet Scrum „Gedränge“. Zwei Mannschaften stehen sich in einer kreisförmigen Aufstellung, dem Gedränge, gegenüber und versuchen gemeinschaftlich, den Gegner daran zu hindern, Raum zu gewinnen (Bild 1.1). Auf Scrum übertragen, steht dieses Bild für den Zusammenhalt, den ein Scrum-Team entwickeln kann, und für das Einhalten weniger Regeln. Rugby ist ein sehr diszipliniertes Spiel. Es geht zwar rau zu, aber die Regeln werden rigide befolgt. Die beiden Organisationstheoretiker Ikujiro Nonaka und Hirotaka Takeuchi greifen in ihrem 1986 erschienenen Artikel „The New New Product Development Game“ den Begriff Scrum auf, um einen neuartigen Produktentwicklungsprozess zu beschreiben [Nonaka, Takeuchi 1986].

Basierend auf der Idee von Nonaka und Takeuchi beschrieben die Autoren Peter DeGrace und Leslie Hulet Stahl 1991 in ihrem Buch „Wicked Problems, Righteous Solutions“, wie man für die Entwicklung von Software statt eines wasserfallartigen Produktentwicklungsprozesses



Bild 1.1 Scrum – Stormers gegen Bulls, Cape Town 16.2.2008. Foto: Boris Gloger

einen „All in one“-Ansatz wählen könnte. Sie erwähnen einen „team approach“, den sie dem Artikel von Nonaka und Takeuchi entlehnen [DeGrace, Hulet Stahl 1990]. Sie schreiben:



„If Scrum were applied to software development, it would go something like this: Suppose you have a software development project to do. For each traditional phase, you can draw from a pool of experienced people. Rather than have several designers to do the design phase and have several coders to do the construction phase, etc., you form a team by carefully selecting one person from each pool.

During a team meeting, you will tell them that they have been carefully chosen to do a project that is very important to the company, country, organization, or whatever. This unsettles them somewhat. You then give them a description of the problem to be solved, the figures for how much it costs in time and money to do similar projects and what the performance figures for similar systems are. Then, after you have gotten them used to the idea that they are special, having been specially chosen and challenged to do an important job, you further unsettle the team by saying that their job is to produce the system in, say, half the time and money and it must have twice the performance of other systems. Next, you say that how they do it is their business. Your business is to support them in getting resources. Then you leave them alone. You stand by to give them advice if you are asked. You get their reports, which come regularly but not as often nor as voluminously as with the Waterfall model.“

Die beiden haben sogar in diesem ersten Versuch, Scrum zu erklären, deutlich gemacht: „The Scrum approach works by reducing the need for lengthy, passive, third-person communication, reporting, and authorization ceremonies. And, it substitutes short active, first- and second-person communication and fewer contacts with ‚external‘ entities. Its basic premise is that if you are committed to the team and the project, and if your boss really trusts you, then you can spend time being productive – not justifying your work.“

[DeGrace, Hulet Stahl 1990]

Faszinierend finde ich an diesem kurzen Ausschnitt aus dem Buch von DeGrace und Hulet Stahl, dass die beiden bereits erkennen, dass Scrum in zwei Umfeldern möglicherweise nicht funktionieren könnte: dem Maintenance-Umfeld und in großen Projekten. Sie haben aber den Grundstein für eine Diskussion gelegt, an deren Ende heute klar ist, dass man Scrum skalieren kann und dass sich mit einer Visualisierungstechnik aus dem Lean Manufacturing auch im Maintenance-Umfeld, durch einen Blend aus Scrum mit Kanban, die Arbeit bestens steuern lässt. Die beiden haben ebenfalls erkannt, wo bei einer Implementierung von Scrum die Probleme liegen:

1. *„As you might imagine, using the Scrum approach would take some courage on the part of management even with the success of the Japanese economy to use as support.“ (...)*
2. *„(...) it would be difficult, and sometimes even impossible, give up the control that it takes to support the Scrum method. In the first place, the government, as a customer, would object since they have a historical need to ‚oversee‘ the spending of tax dollars. In the second place, most companies have stockholders with a need for dividends. The Scrum approach is risky.*

Finally there is no guarantee that the team will not run up against real (rather than institutional) limits which could mean failure of the project.“

Diese Argumente werden auch heute noch gegen Scrum bemüht. Über die Jahre hat sich aber gezeigt, dass sie nicht ins Gewicht fallen. Durch die theoretische Fundierung von Scrum mit mathematischen und logistischen Theorien, wie wir sie durch die Arbeiten von Don Reinertsen kennen (siehe Abschnitte 1.4.2 und 1.4.4), wissen wir auch, wieso sich Scrum gegen diese vermeintlichen Probleme erfolgreich behaupten konnte. Aber sehen wir uns zunächst an, wie das Prozessmodell von Scrum funktioniert. Danach machen wir eine kurze Zeitreise in die Entstehungsgeschichte von Scrum und wir werden uns genauer damit beschäftigen, wie sich die Bedeutung von Scrum in den letzten Jahren verändert hat.

1.1.1 Eine kurze Einführung in die Funktionsweise

Das Prozessmodell von Scrum besteht aus Rollen, Meetings und Artefakten und steckt den Rahmen ab, in dem alle Aktivitäten der Produktentwicklung ablaufen.

Am Anfang der *Discovery-Phase* steht die Person mit einer Produktidee. Häufig ist das ein Kunde oder eine andere Person, die einen Auftrag für eine Produktentwicklung geben kann. Diese Person bespricht ihre Idee mit dem Product Owner, der die Idee – oft auch direkt mit dem Entwicklungsteam – so lange bearbeitet, bis eine *Produktvision* entstanden ist. Die Produktvision enthält die grundlegende Idee für das Produkt. Der Product Owner erarbeitet – entweder alleine oder mithilfe der Teammitglieder – die Produktfunktionalitäten, Product Backlog Items oder auch User Storys genannt. Alle Product Backlog Items werden in eine Liste eingetragen, das *Product Backlog*. Der Product Owner bringt die Product Backlog Items in dieser Liste in eine Reihenfolge, die sich aus dem zu erwartenden finanziellen Gewinn der jeweiligen Funktionalität ergibt. Der gesamte Prozess wird vom ScrumMaster gemanagt. Die drei Rollen Product Owner, ScrumMaster und Entwicklungsteam agieren gemeinsam als das *Scrum-Team*.

Strategische Planung

Jedes Product Backlog Item muss auf seine Größe geschätzt werden. Die *Schätzung* wird von den Teammitgliedern des Entwicklungsteams durchgeführt. Ein Entwicklungsteam besteht aus allen Personen, die notwendig sind, um die Backlog Items in Software zu verwandeln, die ausgeliefert werden kann. Die Teammitglieder schätzen also den Umfang jedes zu liefernden Product Backlog Items und teilen das Ergebnis dem Product Owner mit. Diese initiale Schätzung und das initiale Erarbeiten des Product Backlogs geschieht in der strategischen Planungsphase, d. h. bevor das Team die Arbeit an der Produktentwicklung aufnimmt (Bild 1.2).

Das Product Backlog ist nun geschätzt. Alle Teammitglieder haben eine Vorstellung davon, wie das gewünschte Produkt aussehen soll, und der Product Owner gewinnt einen ersten Eindruck, welche Kosten auf ihn zukommen. Nun nimmt das Entwicklungsteam die Arbeit auf. In Scrum wird in klar abgegrenzten zeitlichen Intervallen, den *Sprints*, gearbeitet. Am Ende eines Sprints muss das Team *Produktinkremente* liefern, die potenziell ausgeliefert werden könnten. In der Scrum-Terminologie gibt es für dieses entstandene Produktinkrement mehrere Bezeichnungen: Potential Shippable Business Functionality, Potential Shippable Code.

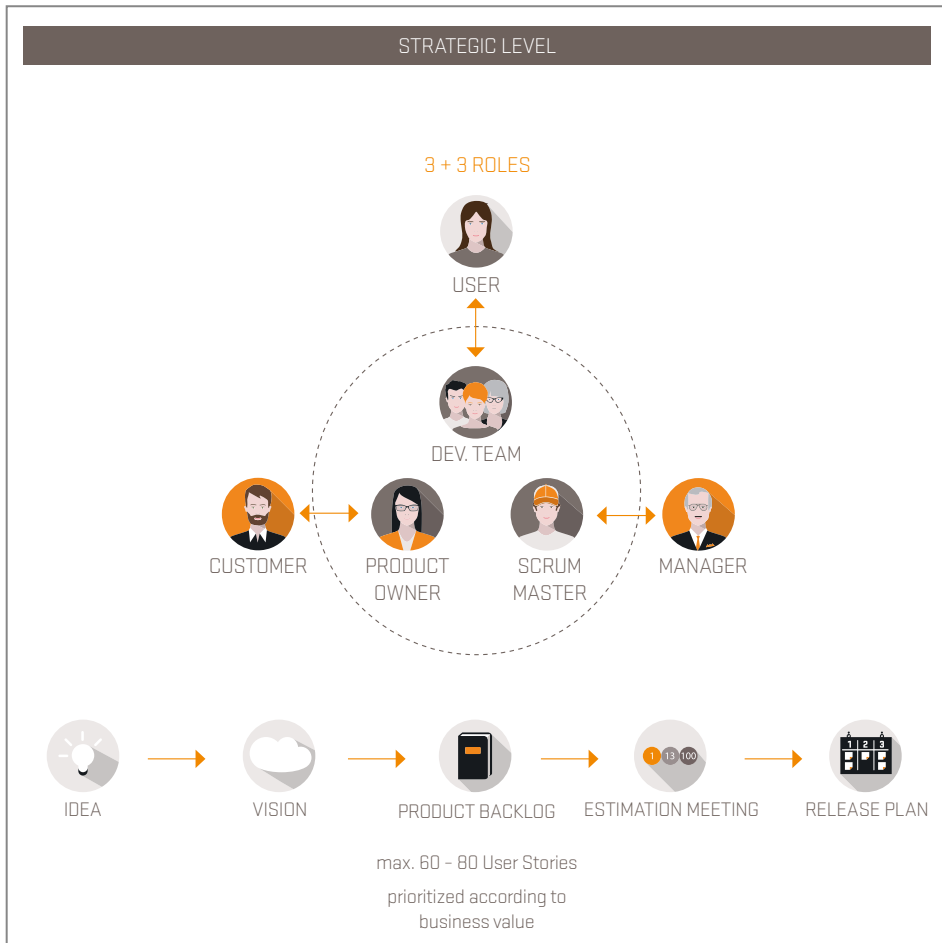


Bild 1.2 Der Scrum Flow auf strategischer Ebene

Taktische Planung

Jeder Sprint beginnt mit der taktischen Planung, die in zwei Phasen unterteilt wird (Bild 1.3). Das erste Planungsmeeting ist das *Sprint Planning Meeting 1*. Hier besprechen der Product Owner, das Entwicklungsteam, das Management und die Anwender des Produkts die Ziele des Sprints, das Sprint Goal. Sie legen gemeinsam fest, wie viele und welche Product Backlog Items (User Stories) das Team in diesem Sprint liefern will. Dieses Meeting ist mit einem Briefing oder einem Anforderungsworkshop vergleichbar. Alle Beteiligten haben nach diesem Workshop eine klare Vorstellung davon, was in diesem Sprint geschehen und geliefert werden soll.

Die Summe der vereinbarten Product Backlog Items (User Stories), die das Entwicklungsteam liefern will, wird als Selected Product Backlog bezeichnet. Wenn allen Beteiligten klar ist, was das Selected Product Backlog umfasst, und sich das Team und der Product Owner darauf geeinigt haben, wird die Summe der User Storys gemeinsam mit dem Sprint Goal als gemeinsames Commitment (Versprechen) des Product Owners und des Entwicklungsteams festgehalten.

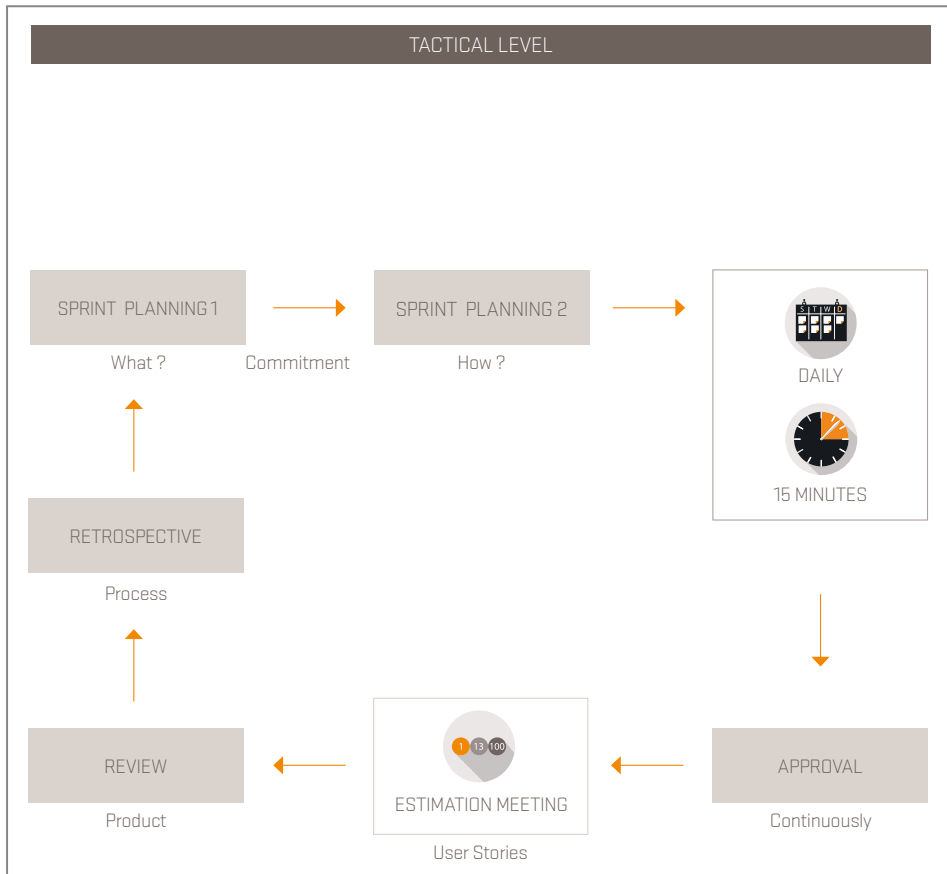


Bild 1.3 Der Scrum Flow auf taktischer Ebene

Der darauf folgende Workshop, das *Sprint Planning Meeting 2*, gibt dem Team die Gelegenheit, darüber zu diskutieren, wie es die Anforderungen an die Backlog Items aus dem Sprint Planning Meeting 1 umsetzen will. Dieser Workshop ist mit einem Design-Workshop vergleichbar. Das Team entwickelt erste Vorstellungen davon, wie es die Anforderungen umsetzen will. Dazu erstellen die Teammitglieder erste Designskizzen und legen die Architektur fest. Häufig überprüfen die Mitglieder des Entwicklungsteams, was schon vorhanden ist, zum Beispiel wie die Codebasis aussieht. Da in diesem Prozess für alle Teammitglieder deutlich wird, was zu tun ist, entsteht auf diese Weise eine Liste aus Aufgaben, das Sprint Backlog. Das Sprint Backlog ist eine Liste aller Aufgaben, die das Team ausführen muss, um die geforderten Product Backlog Items umzusetzen.

Nun beginnt die Umsetzung, also das Entwickeln des Produkts. Die Teammitglieder stimmen im sogenannten *Daily Scrum* ihre Aufgaben täglich untereinander ab. Das Daily Scrum ist das Meeting, in dem alle Teammitglieder bekannt geben:

- Was habe ich seit dem letzten Daily Scrum erreicht?
- Was will ich bis zum nächsten Daily Scrum erreichen?

- Welche Impediments (Hindernisse) stehen mir dabei im Weg?
- Wie kann ich einem Kollegen helfen, damit er möglicherweise schneller fertig wird?

In diesem Meeting bestimmen die Teammitglieder selbst, welche Aufgaben aus dem Sprint Backlog sie abarbeiten wollen. Während dieses Meetings wird auch festgehalten, ob neue Aufgaben hinzugekommen sind oder ob Aufgaben aufgeteilt werden, um sie besser handhabbar zu machen.

Während des Sprints bereiten die Teammitglieder gemeinsam mit dem Product Owner die nächsten Sprints vor. Das Product Backlog wird aktualisiert, für neue Backlog Items werden Schätzungen erstellt und frühere Schätzungen gegebenenfalls aktualisiert. Formal finden diese Arbeiten im *Estimation Meeting* statt, das den Input für den Releaseplan des Projekts liefert. Am Ende des Sprints präsentiert das Team die fertiggestellte Funktionalität in einem weiteren Workshop, dem *Sprint Review*. Hier wird der Fortschritt des Projekts anhand des „Potential Shippable Product Increment“ demonstriert. Diese Demonstration zeigt deutlich, wo die Produktentwicklung steht, und führt in der Regel zu neuen Ideen. Sollte genug Funktionalität für die Freigabe des Produkts an die Produktion vorhanden sein, werden im Anschluss die entsprechenden Maßnahmen gesetzt, damit der Release stattfinden kann. Nach dem Sprint Review führen die Teammitglieder gemeinsam die *Sprint Retrospektive* durch. In diesem Meeting optimiert das Team die eigenen Arbeitsprozesse, indem es sie konsequent hinterfragt.

Dieser Scrum-Prozess wird durch den ScrumMaster initiiert und am Leben erhalten. Der ScrumMaster steht sozusagen außerhalb des Prozesses, aber dazu später mehr.

1.1.2 Rollen, Meetings, Artefakte

Wir unterscheiden in Scrum sechs Rollen, sechs Meetings und acht Artefakte. Durch ihr Zusammenspiel können sich hochproduktive Projektteams formieren. Hier zunächst die Rollen, Meetings und Artefakte im Überblick, damit wir in weiterer Folge ein gemeinsames Bild davon haben.

Der Product Owner – der Visionär

Der Product Owner plant und lenkt die Produktentwicklung. Er steuert das Team, indem er eine deutliche Vision vorgibt, klare und eindeutige Rahmenbedingungen definiert und festhält. Er ist verantwortlich dafür, dass das Produkt den gewünschten Return on Investment bringt, indem er vorgibt, in welcher Reihenfolge das Entwicklungsteam die gewünschten Funktionalitäten liefern soll. Er sorgt dafür, dass die Projektergebnisse den finanziellen Aufwand für das Projekt rechtfertigen. Dazu arbeitet er täglich mit dem Entwicklungsteam, trifft zeitnah die notwendigen Entscheidungen und arbeitet kontinuierlich am Product Backlog und dem Releaseplan.

Das Entwicklungsteam – die Lieferanten

Das Entwicklungsteam arbeitet so dicht wie möglich am User und designt und entwickelt das Produkt. Ein Entwicklungsteam besteht aus den Personen, die für die Lieferung des Produkts erforderlich sind. Diese sogenannten crossfunktionalen und multidisziplinären Teams sind so zusammengestellt, dass sie ohne den Einfluss der Organisation das Produkt liefern können. Ein Entwicklungsteam managt seine Angelegenheiten selbst. Es ist autorisiert, alles

Zielführende zu tun, um das angestrebte Ergebnis zu erreichen. Gleichzeitig sollte es die Standards und Prozesse der Organisation einhalten. Das Entwicklungsteam steuert selbst die Arbeitsmenge, die es bewältigen kann. Dafür trägt es die Verantwortung für die Qualität der Lieferung und arbeitet so professionell wie möglich. Es ist immer am neuesten Stand der jeweiligen Technologie und liefert State of the Art.

Der ScrumMaster – der Change Agent

Der ScrumMaster führt das Scrum-Team. Er ist dafür verantwortlich, dass die Produktivität des Scrum-Teams kontinuierlich steigt und arbeitet daran, dass alle Schwierigkeiten, Blockaden und Probleme, die das Team aufhalten, gelöst werden. Zwar ist der ScrumMaster nicht weisungsbefugt, er sorgt aber dafür, dass der Scrum-Prozess eingehalten wird. Seine Hauptaufgabe besteht darin, alle am Projekt beteiligten Personen so zu schulen, dass sie ihre Rolle verstehen und ausüben können.

Der Kunde – der Finanzierer

Der Kunde ist Anforderer des Projekts, er kauft es oder hat es in Auftrag gegeben. Typischerweise sind das die Executive Manager in Organisationen, die Softwareentwicklung bei externen Unternehmen einkaufen. In einem internen Projektentwicklungsteam ist der Budgetverantwortliche in der Rolle des Kunden.

Der Anwender – der Nutzer

Der Anwender des Produkts ist eine wesentliche Informationsquelle für das Scrum-Team. Er ist es, der später die „Usable Software“ benutzen wird. Daher bezieht das Scrum-Team den Anwender in die Produktentwicklung mit ein, so gut es geht. Beim Sprint Planning definiert er gemeinsam mit dem Product Owner die Anforderungen. Später wird er als Anwender mit dem Team daran arbeiten, die Anwendung nutzbar zu machen.

Der Manager – der Bereitsteller

Das Management stellt die Ressourcen und die Richtlinien innerhalb einer Organisation bereit. Es schafft den Rahmen, in dem sich das Entwicklungsteam, der Product Owner und der ScrumMaster bewegen. Oft löst es die vom ScrumMaster identifizierten Probleme.

Die Meetings

Sprint Planning Meeting 1 – Anforderungen klären

In diesem ersten Meeting eines Sprints sind der Product Owner, das Entwicklungsteam, das Management, der Anwender und der ScrumMaster anwesend. Der Product Owner erläutert die Product Backlog Items und definiert gemeinsam mit den Teammitgliedern und dem Management das Ziel für den anstehenden Sprint. Dann werden die Backlog Items ausgewählt, die zu diesem Ziel passen und die das Team schaffen kann. So entsteht das Selected Product Backlog. Wichtig: Das Entwicklungsteam alleine bestimmt, wie viele Backlog Items es auswählt.

Sprint Planning Meeting 2 – Design und Planung

Im Sprint Planning Meeting 2 plant das Entwicklungsteam, wie es das im Sprint Planning Meeting 1 vereinbarte Ziel erreichen will. Dazu beraten die Teammitglieder, wie die Applikation aufgebaut sein soll, welche Architektur gewählt werden muss, welche Interfaces geschrieben werden sollen, ob bereits Test Cases erstellt und geschrieben werden sollen, kurz: Sie besprechen detailliert, was getan werden muss. Am Ende dieses Meetings liegt eine Liste aller notwendigen Aufgaben vor: das Sprint Backlog.

Daily Scrum – Koordination und Feedback

Jeden Tag treffen sich die Mitglieder des Entwicklungsteams zur gleichen Zeit am selben Ort für 15 Minuten zu einem vom ScrumMaster moderierten Meeting. In diesem Meeting sprechen sie ab, wer an diesem Tag welche Aufgabe übernimmt. Dabei wählen die Teammitglieder selbst die Aufgabe aus, die sie als Nächstes übernehmen wollen. Sie informieren den ScrumMaster über Blockaden und Probleme, damit dieser sie so schnell wie möglich zur Lösung bringen kann.

Estimation Meeting – planen und schätzen

Der Product Owner und die Mitglieder des Entwicklungsteams aktualisieren mindestens einmal im Sprint das Product Backlog. Dabei werden Backlog Items mit neuen Schätzungen versehen und neue Backlog Items in das Product Backlog aufgenommen. Gleichzeitig wird die Reihenfolge der Backlog Items unter Berücksichtigung der neuen Informationen angepasst. Dieses Meeting dient auch dazu, den Releaseplan des Projekts zu aktualisieren und zu vervollständigen.

Sprint Review – Resultate präsentieren

Am Ende des Sprints präsentiert das Entwicklungsteam die erarbeiteten Funktionalitäten. Das Team zeigt nur die Funktionalitäten, die so weit sind, dass sie sofort produktiv gesetzt werden könnten. Nicht getestete oder instabile Funktionalitäten werden nicht gezeigt und gelten als nicht geliefert.

Sprint Retrospektive – sich ständig verbessern

Die Sprint Retrospektive ermöglicht dem Team das systematische Lernen. Hier wird analysiert, welche Arbeitsprozesse verbessert werden müssen, damit das Team effektiver arbeiten kann. Die Resultate aus der Retrospektive werden im Impediment Backlog festgehalten und lassen sich so als Verbesserungsvorschläge in das Sprint Planning einbringen.

Die Artefakte – Resultate des Scrum-Prozesses

Die Produktvision

Eine Vision ist ein klares, deutliches, attraktives und emotionalisierendes Bild des angestrebten Produkts.

Das Product Backlog Item (User Story)

In Scrum werden die zu liefernden Funktionalitäten, die in einem Product Backlog aufgelistet sind, als Product Backlog Items bezeichnet. Diese werden meist in Form von sogenannten User Storys notiert.

Das Product Backlog

Das Product Backlog ist eine Liste von Product Backlog Items (Storys). Die Storys sind in diesem Backlog nach ihrer Wichtigkeit sortiert.

Das Sprint Goal

Der Product Owner legt gemeinsam mit dem Entwicklungsteam das Sprint Goal, das Ziel des Sprints fest.

Die Aufgaben

Mit Aufgaben bezeichnen wir alles, was getan werden muss, um das Ziel des Sprints zu erreichen und die geforderten Funktionalitäten zu entwickeln.

Der Releaseplan

Der Releaseplan zeigt an, in welchem Sprint welche Backlog Items vom Entwicklungsteam geliefert werden können. Er ist kein Instrument, um einem Team zu sagen, was es wann zu liefern hat, sondern er zeigt nur an, wann das entsprechende Backlog Item vom Team zu erwarten ist. Er ist also ein Informationsinstrument und kein Planungsinstrument im klassischen Sinn.

Das Impediment Backlog

Das Impediment Backlog ist die Liste aller Hindernisse, die einem Team aus dem Weg geräumt werden müssen, damit es produktiver werden kann.

Das Produktinkrement – Usable Software

Das Entwicklungsteam strebt an, am Ende eines Sprints etwas zu liefern, das man präsentieren kann. Die Funktionalitäten, die am Ende des Sprints präsentiert werden, müssen tatsächlich in einem Zustand sein, in dem sie ausgeliefert und verwendet werden können.

■ 1.2 Ein Begriff – viele Einsatzmöglichkeiten

Ist Scrum die bessere Projektmanagementmethode? Wofür lässt sich Scrum einsetzen? Behalten wir im Hinterkopf, dass Scrum in erster Linie ein Managementframework ist, das teamzentriert dazu führt, dass Organisationen kundenorientiert und iterativ Werte (also in der Regel Produkte) schaffen können.⁴

Durch seine Eigenschaft als Managementframework und Geisteshaltung kann Scrum auf verschiedenen Ebenen zum Erfolg führen und die Anwendungsgeschichte von Scrum zeigt es ganz deutlich: Kleine Teams können damit genauso gemanagt werden wie große Projekte mit Hunderten Mitgliedern und sogar ganze Unternehmen.

Vor allem wegen seines Ursprungs wurde Scrum lange Zeit lediglich als Methode gesehen, um Softwareentwicklungsteams produktiver zu machen. Ken Schwaber, einer der Begründer von Scrum, war selbst Softwareentwickler und hat anschließend Unternehmen dabei beraten, professionelle Softwareentwicklungsprozesse zu etablieren. Im Laufe der Entstehung agiler Ansätze wurde Scrum auch in diese Ecke gerückt und nach wie vor wird es oft als eine *Softwareentwicklungsmethode* gesehen. Allerdings ist diese Sichtweise nicht korrekt, denn Scrum erklärt einem Entwicklungsteam nicht, wie es Software entwickeln soll. Es zeigt einem Team nur, wie es sich organisieren soll.

Scrum als Projektmanagementverfahren

Der allgemeingültige Aspekt der Teamorganisation führte schnell dazu, dass Scrum erfolgreich bei Softwareentwicklungsprojekten als Projektmanagementverfahren eingesetzt wurde. Mit Scrum wurden diese Projekte deutlich effektiver umgesetzt als mit herkömmlichen Methoden. Zunächst wurde Scrum für kleinere Projektteams eingesetzt, später wurden

⁴ Stephen Denning, der ja nicht aus der Scrum Community, sondern aus dem klassischen Management kommt, hebt vor allem diesen Aspekt hervor [Denning 2010].

Verfahren entwickelt, um auch größere Projektteams auf diese Art und Weise zu steuern. In seinem Buch „Agile Project Management with Scrum“ beschreibt Ken Schwaber bereits 2005, wie man Scrum beim Führen von Projekten einbringt. In ihrem Buch „Software in 30 Days“ beschreiben Ken Schwaber und Jeff Sutherland am Beispiel des Projekts „Sentinel“ des FBI, wie mithilfe von Scrum die entscheidende Phase des Projekts bewerkstelligt werden konnte [Schwaber, Sutherland 2012].

Scrum wird gerne als Projektmanagementmethode eingesetzt, weil es Projektteams dabei hilft, konzentriert und fokussiert zu arbeiten. Mit einem Product Owner als „Projektleiter“ und einem Scrum-Master als „Teamleiter“ haben es Unternehmen anscheinend auch sehr einfach, die alten Rollen unter einem neuen Namen wieder zu besetzen. Allerdings haben Menschen, die in traditionellen Denkweisen verhaftet sind, Schwierigkeiten zu verstehen, dass ein Backlog kein Projektplan ist und sich ein Releaseplan in Scrum kontinuierlich ändert. Die in Scrum konsequent durchgeführte rollierende Planung (Rolling Wave Approach) ist für viele Projektmanager nicht so einfach umzusetzen, wie es den Anschein hat. Die Überlegungen aus dem klassischen Denken, dass man zu Beginn des Projekts wissen soll, wann es fertig ist, widerstreben der Vorstellung, dass man erst während der Laufzeit des Projekts das konkrete Ziel gemeinsam (er-)findet.

Ich schlage Ihnen daher folgenden Gedanken vor: Stellen Sie sich einfach vor, Sie führen in Ihrem traditionell geplanten Projekt mehrere Phasen durch. Sie schneiden das Gesamtprojekt in Abschnitte und überlegen, welcher Abschnitt zwar nicht vollständig, aber grundsätzlich lieferbar wäre. Dafür nehmen Sie sich etwas Zeit – sagen wir drei bis sechs Monate. In dieser Zeit führen Sie mithilfe von Scrum mehrere kleine Iterationen durch, erzeugen nach außen für die Organisation aber ein ganz normales Projekt mit der ersten Phase und liefern diese ersten Teile des Projekts. Dann stoppen Sie die übrigen Phasen, gehen übergangslos in den Wartungsmodus Ihres Projekts über und liefern so oft Sie können (in kurzen Sprints) kleinteilig neue Funktionalitäten.

Scrum als Produktentwicklungsmethode

Die Sichtweise auf Scrum verändert sich vollkommen, wenn man die Problematik vom Ende her zu denken beginnt. Im Grunde geht es doch darum, ein Produkt oder einen Service aus Kundensicht iterativ zu entwickeln. In Wahrheit handelt es sich bei Scrum also um eine Produktentwicklungsmethode und nicht um eine Projektmanagementmethode. Sicher könnte man nun argumentieren, dass mithilfe eines Projekts ein Produkt entwickelt wird. Aber versuchen Sie doch mal, den Gedanken des Projekts aufzugeben, und versetzen Sie sich in die Lage, dass Sie einen kontinuierlichen Strom an Produktteilen liefern müssen. Diese Produktteile sind die Ideen für weitere Produktteile, die Sie in eine Liste schreiben, und nun arbeiten Sie diese Liste Teil für Teil ab. Spinnen Sie den Gedanken konsequent weiter, dann hört das Projekt nicht auf, die Entwicklung des Produkts geht also bis in alle Ewigkeit weiter. In diesem Fall hat es gar keinen Sinn, darüber nachzudenken, wann man fertig ist. Denn das Produkt lebt. Es muss sich immer weiterentwickeln. Mit Scrum wird ein Produkt von der Idee bis zu seiner Eliminierung aus dem Sortiment betreut und daher wird mit Scrum ein gesamter Produktlebenszyklus gemanagt.

Scrum für das sehr kleine Team

In vielen Unternehmen findet man sehr kleine Teams, in denen maximal zwei Personen über Monate, manchmal sogar über Jahre an denselben Projekten und Produkten arbeiten. Natürlich ist es sinnlos, für diese Teams einen Product Owner und einen ScrumMaster abzustellen. Das heißt aber nicht, dass man Scrum deshalb nicht einsetzen kann. Der Begriff des Projekts muss in diesem Fall aber tatsächlich aufgelöst werden und sollte durch „Entwicklungsabteilung“ oder „Entwicklungsteam“ ersetzt werden. Alle Miniprojektteams werden vereint und es entsteht ein einziges großes Scrum-Team (Bild 1.4). Dieses wird von einem Product Owner (z. B. Abteilungsleiter ohne disziplinarische Verantwortung) mit Arbeit versorgt. Wie wir später noch sehen werden, schafft dieses Verfahren sogar eine höhere Produktivität.

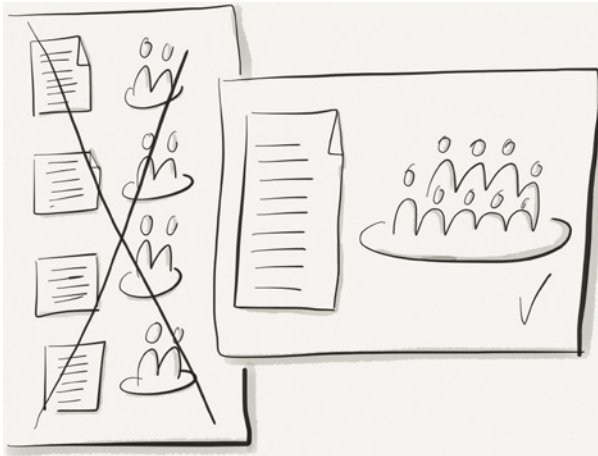


Bild 1.4

Zusammenfassung kleiner Teams in ein großes Scrum-Team

Scrum für das große Produkt – Skalierung

Die Selbstorganisation von Teams kann theoretisch auch in einem Unternehmen mit Tausenden Mitarbeitern stattfinden. Es gibt heute sichere Verfahren, um mit Scrum und Visualisierungstechniken auf diese Größenordnungen zu skalieren. Gepaart mit einer möglichst entkoppelnden Architektur eignet sich Scrum durchaus dafür, ein großes Produkt zu entwickeln. In diesem Fall werden die einfachen Regeln von Scrum dazu genutzt, größere Zusammenhänge auszusteuern. Problematisch ist dabei meistens die Tatsache, dass es erst wenige Leute gibt, die damit Erfahrung haben. Im Grunde ist das Wissen aber bereits seit einigen Jahrzehnten vorhanden: Man kann den Vergleich zu Industriezweigen ziehen, in denen Hunderte, manchmal sogar Tausende Menschen gemeinsam an einem Produkt arbeiten, etwa im Schiffsbau.

Die Wissensprodukte großer Softwareentwicklungsprojekte haben dabei ein Problem, das greifbare Produkte nicht haben. Nehmen wir den Frankfurter Flughafen als Beispiel: Diesen Flughafen hätte man in seiner heutigen Form in den 1970er-Jahren nie und nimmer im Rahmen eines einzigen großen Projekts erbauen können. Ich bin in dieser Region aufgewachsen und seit ich ihn als Kind zum ersten Mal bewusst zur Kenntnis genommen habe, wird an diesem Flughafen gebaut. Heute, als erwachsener Vielflieger, lande ich fast jede Woche in Frankfurt – und der Flughafen wird noch immer umgebaut und umgebaut und umgebaut.

Sogar neue Start- und Landebahnen kommen dazu und das alles bei Vollbetrieb. Würde man einen Flughafen dieser Dimension sofort aus dem Boden stampfen wollen, weil man sich das langsame Wachstum ersparen will, würde man nicht fertig werden. Softwareentwicklungsprojekte sind mit dem Umbau des Frankfurter Flughafens vergleichbar. Wer heute Facebook, Google, MS-Word oder Adobe Photoshop nachbauen will, braucht dazu Jahre. Diese Applikationen enthalten so viel Arbeit, dass das Nachprogrammieren so gut wie unmöglich ist.

Scrum für die Steuerung des Produktmanagements

Scrum ist hervorragend dazu geeignet, im Produktmanagement eingesetzt zu werden. Beispielsweise kann eine Produktmanagementabteilung mithilfe von Scrum organisiert werden. Allerdings reicht Scrum allein dafür nicht aus. In diesem Fall baut Scrum eine neue Struktur des Linienmanagements in der Produktmanagementorganisation auf. Das wirkt sich sogar so weit aus, dass Teams zur Gänze dem Produktmanagement zugeordnet werden und die Produktmanager (Product Owner) selbstverständlich einen obersten Produktmanager haben, der die strategische Richtung vorgibt. Eine Mischung aus den Ideen von Scrum und den Ansätzen, die sich bei Marty Cagan finden, führt hier zur Lösung: „It is the primary job of every manager to build and develop the capabilities of his or her team, but this is especially true for the role of the Director of Product Management because of the high impact nature of the Product Manager position.“ [Cagan 2008] Weiter schreibt er: „Nobody is more responsible or more accountable for the suite of products a company offers than the head of the product management organisation. This person decides what products to pursue, and then closely reviews the strategy and the progress for each product.“ Mithilfe von Scrum lassen sich die Produktmanager als Team von Product Ownern führen und es ist möglich, den Fortschritt jedes Produkts auf „Sprint by Sprint“-Basis zu reviewen.

Scrum für Business Intelligence

Immer mehr Unternehmen – zum Beispiel Cundus, Woodmark, aber auch SAP – setzen bei der Einführung von Business-Intelligence-Systemen für ihre Kunden auf Scrum. Auch bei diesen Projektarten zeigt sich, dass die überbordende Komplexität in den Organisationen dazu zwingt, in kleineren Paketen vorzugehen. Für Berater wird Scrum in diesen Umgebungen entweder zur Chance oder zur Bedrohung. Die klassischen Methoden der Einführung von Data-Warehouse-Lösungen waren geradezu prädestiniert für den Wasserfallansatz. Oft überfordert es das Denken der Projektmanager, einen Report nach dem anderen zu erzeugen und eine gesuchte Information nach der anderen anzugehen.

Scrum als Methode der Organisationsentwicklung

In „The Enterprise and Scrum“ zeigt Ken Schwaber, wie man mit Scrum sogar eine Organisation verändern kann [Schwaber 2007]. Scrum als Methode der Organisationsentwicklung hat sich seitdem weiterentwickelt und immer mehr Elemente aus den klassischen Organisationsentwicklungsschulen wurden in einen Scrum-Rahmen gegossen. Das Paradigma der Organisationsentwicklung mit Scrum besteht darin, eine Organisation entlang ihrer Produkte zu entwickeln. Die Organisation wird in ihren Strukturen gezielt auf das ausgerichtet, was sie ausmacht: ihr ganz eigenes, unverwechselbares Produkt.

Scrum als Managementansatz

Was ist eigentlich Management? Ist es „ein Beruf, eine Praxis, eine Disziplin?“, wie Fredmund Malik sagt, „ist [es] die bewegende Kraft, überall wo es darum geht, durch ein arbeitsteiliges Zusammenwirken vieler Menschen gemeinsam etwas zu erreichen“? [Malik 2007] Oder ist es viel mehr? Peter Druckers Standardwerk zum Management ist rund 840 Seiten dick. Seine Auffassungen fußen auf den Theorien von Max Weber und er war der Meinung, dass Management eine Profession oder ein Handwerk sei, das inhaltlich nicht zwangsläufig etwas mit der Arbeit zu tun haben müsse, die in einem Unternehmen verrichtet wird [Drucker 1985]. Die meiner Meinung nach derzeit modernste Auffassung ist jene von Thomas J. DeLong. Sie ist möglicherweise die erste, die sich auch mit den Ideen zum Management 2.0 von Gary Hamel vereinen lässt: Nach Meinung von DeLong ist es die Aufgabe der Führung in einem Unternehmen, die Richtung vorzugeben, die Mitarbeiter dazu zu bringen, sich freiwillig in diese Richtung zu bewegen, dafür zu sorgen, dass die Ausführung der Arbeiten auch wirklich geschieht, und als Manager selbst ein Beispiel dafür zu sein, wie diese Aufgaben erfüllt werden können [DeLong et al. 2007].

Scrum stellt Managern genau diesen Rahmen zur Verfügung, mit dem sie ihre Führungsaufgaben umsetzen können. Scrum bietet einen Weg, Mitarbeitern eine Richtung vorzugeben, sie eigenständig arbeiten zu lassen und ihnen dabei die finanzielle Verantwortung zu überlassen. Es geht also nicht nur darum, Menschen zu etwas zu befähigen, sondern – so wie es Tom Peters fordert – ihre Jobs in ein eigenständiges Business zu verändern [Peters 1994]. Insofern lässt sich Scrum als Managementansatz für ganze Organisationen verstehen, der mithilfe von Pull-Prinzipien, Lean-Management-Ideen und den systemischen Prinzipien der Selbstorganisation Menschen dabei hilft, in immer komplexeren Umfeldern miteinander zu interagieren.

Scrum in der Organisation und die Rahmenbedingungen

Noch immer gilt: Nicht die ganze Organisation muss Scrum machen, damit man als Einzelner beginnen kann, Scrum für sein Team einzusetzen. Je größer allerdings eine Organisation ist, desto wichtiger wird es für diese Organisation, dass möglichst viele Abteilungen wissen, was Scrum ist, und sich auch entsprechend ausrichten – egal, ob es die HR-Abteilung, der Einkauf, der Verkauf, das Projektmanagement-Office, der Betrieb oder das Marketing ist. Sie alle werden früher oder später davon betroffen sein, wenn die Produktentwicklungsorganisation beginnt, sich mit Scrum zu organisieren. Diese Abteilungen haben natürlich die Möglichkeit, mit Ablehnung und Widerstand zu reagieren. Sie können aber auch versuchen zu verstehen, wie mit Scrum die eigene Arbeit leichter und bedeutungsvoller werden kann.

■ 1.3 Scrum – eine Bewegung entsteht



Juli 2003, Salt Lake City. Es war Sonntagabend, ich ging einen langen Flur hinunter. Musik war zu hören. Erwartungsvoll betrat ich den Saal, in dem die Ice-Breaker-Party stattfand. Etwa 100 Menschen waren bereits da und schauten sich suchend um. Alle waren gekommen, um die erste „Agile Development Conference“ zu besuchen – Alistair Cockburn, Gründer der Agile Alliance, hatte eingeladen. Aus Japan, Brasilien, den USA, England, Deutschland, Österreich, Frankreich und vielen anderen Ländern waren wir angereist, um mehr über agile Softwareentwicklung zu erfahren.

Ich stand ein wenig verloren da, als mich eine junge Frau ansprach. Sie trug ein T-Shirt mit der Aufschrift „Agile Development Conference“ und erzählte, sie sei eine der vielen Studentinnen, die Alistair engagiert hatte, um die Besucher der Party in Gespräche zu bringen. Ich war auf der besten Ice-Breaker-Party gelandet, die ich jemals besucht hatte. Die Idee, Verbindungen und Beziehungen innerhalb der agilen Gemeinschaft aufzubauen, ist das, was mir von dieser Konferenz am deutlichsten in Erinnerung geblieben ist. Jeder unterhielt sich mit jedem, weil jeder jeden miteinander bekannt machte. Alle Teilnehmer dieser Konferenz kannten sich am Ende. Jeder hatte einmal mit Ken Schwaber, Alistair Cockburn, Ron Jeffries oder Tim Lister geredet. Diese Konferenz zeigte eindrucklich, worum es bei agiler Softwareentwicklung geht: um das gemeinsame Arbeiten, um innovative Resultate zu erzielen. Wir alle gehörten damals zu einer Gruppe von Menschen, die gerade dabei waren, einen leisen Aufstand zu planen und durchzuführen.

Dabei war diese Bewegung erst zwei Jahre zuvor sichtbar geworden. Alistair Cockburn hatte damals die Begründer der agilen Softwareentwicklungsmethoden nach Snowbird, Utah, eingeladen. Dort war 2001 das Agile Manifesto entstanden.

Mehr als zehn Jahre später, Europa. Die Gruppe hat ihr Vorhaben erfolgreich umgesetzt. Immer mehr Unternehmen setzen Scrum und andere agile Entwicklungspraktiken ein. In Europa sind die Ideen der agilen Bewegung in den Softwareentwicklungsabteilungen angekommen. Die Menschen, die damals auf dieser Konferenz waren – Ken Schwaber, Alistair Cockburn, Jim Highsmith, Robert C. Martin, Ron Jeffries, Luke Hohmann, Diana Larsen, Esther Derby, Rachel Davis, Mary Lyn Mans und viele andere – sind heute bekannte Gesichter in der Agile Community.

Das Agile Manifest ist für viele Menschen, die agile Softwareentwicklung betreiben, zum Gradmesser ihrer Aktivitäten geworden. Die zwölf darauf aufbauenden Prinzipien waren nur der vorläufige Endpunkt einer Entwicklung, die etwa zehn Jahre zuvor begonnen hatte und im Falle von Scrum durch Jeff Sutherland und Ken Schwaber geprägt wurde.