

# Scrum Think big

Scrum für wirklich große Projekte, viele Teams und viele Kulturen

Bearbeitet von  
Boris Gloer

1. Auflage 2017. Buch. 238 S.  
ISBN 978 3 446 44634 2  
Format (B x L): 17,8 x 24,7 cm  
Gewicht: 568 g

[Weitere Fachgebiete > EDV, Informatik > Software Engineering > Agile Softwareentwicklung](#)

schnell und portofrei erhältlich bei



Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

# HANSER



Leseprobe

Boris Glöger

Scrum Think big

Scrum für wirklich große Projekte, viele Teams und viele Kulturen

ISBN (Buch): 978-3-446-44634-2

ISBN (E-Book): 978-3-446-44813-1

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44634-2>

sowie im Buchhandel.

# Inhalt

<b>Vorwort .....</b>	<b>IX</b>
<b>Über den Autor .....</b>	<b>XIII</b>
<b>1 Die Umfeldbedingungen des Skalierens .....</b>	<b>1</b>
1.1 Hyperspezialisierung .....	5
1.2 Digitalisierung .....	9
1.3 Arbeitsschutzgesetze .....	13
1.4 Professionalität und neue Skills .....	15
1.5 Produktentwicklungszyklen und Projektmanagement .....	17
1.6 Bürokratie und Kontrolle .....	26
1.7 Die Netzwerkorganisation .....	32
1.8 Kapitelausblick .....	34
<b>2 Architektur .....</b>	<b>37</b>
2.1 Architektur als Ergebnis der Kommunikationsstruktur .....	38
2.1.1 Schneller ist besser .....	41
2.1.2 Entkopplung .....	44
2.2 Was ist eine agile Produktarchitektur? .....	47
2.2.1 Microservices – Grundlage flexibler Architektur .....	48
2.2.2 Redundanz und Durchfluss gewährleisten .....	50
2.2.3 Die Einheitlichkeit der Produktarchitektur .....	57
2.2.4 Technologien und Architektur .....	60
2.2.5 Betrieb und Architektur .....	61
<b>3 Infrastruktur .....</b>	<b>63</b>
3.1 Integration ist alles .....	63
3.2 Räume – bauliche Infrastruktur .....	68
3.2.1 Das große Projekt in einem Raum .....	69
3.2.2 Flipcharts, Stifte und Haftnotizen .....	72
3.3 Kommunikationstools .....	73
3.4 Entwicklungstools .....	81

3.5	Zusammenarbeit mit Lieferanten .....	88
3.6	Richtlinien und Policies .....	92
<b>4</b>	<b>Skills und Professionalität .....</b>	<b>97</b>
4.1	Die Skills des Einzelnen .....	100
4.1.1	Selbstmanagement .....	101
4.1.2	Wissen über die Theory of Constraints .....	108
4.1.3	Wissen über neue Formen der Produktentwicklung .....	108
4.1.4	Die Rollen mit den richtigen Fähigkeiten ausfüllen .....	111
	Die Skills des ScrumMasters .....	111
	Die Skills des Product Owners .....	113
4.2	Die Skills des Entwicklungsteams .....	116
4.3	Die Skills des Managements .....	119
<b>5</b>	<b>Produktentwicklung .....</b>	<b>125</b>
5.1	Agiles Anforderungsmanagement .....	128
5.2	Design Thinking .....	130
5.3	Ein agiler Produktentwicklungsprozess .....	134
<b>6</b>	<b>Good Practices für das skalierte Scrum-Projekt .....</b>	<b>139</b>
6.1	Mythos zentrale Steuerung .....	140
6.1.1	Steering Committee und Projektmanagement-Office .....	141
6.1.2	Das richtige Produkt richtig liefern .....	143
	Vertrauen und Erfolg: Im Gehirn des Managers .....	144
6.1.3	Scrum-Teams: gemeinsam für den Return on Investment verantwortlich .....	147
6.2	Theory of Constraints .....	149
6.3	Die Skalierungstoolbox .....	152
6.3.1	Die Meetings .....	152
	Das skalierte Daily Scrum oder Scrum of Scrums .....	154
	Das ScrumMaster Daily .....	159
	Das Product Owner Daily .....	159
	Das ScrumMaster Weekly .....	160
	Das Product Owner Weekly oder Business Value Estimation Meeting ..	160
	Das skalierte Estimation Meeting .....	161
6.3.2	Rollen und Teams .....	161
	Das ScrumMaster-Team .....	161
	Das Product-Owner-Team .....	162
	Die Gilden .....	163
	Support-Teams .....	164
6.3.3	Artefakte .....	167
6.4	Agiles Portfoliomanagement .....	169
6.4.1	Wert und Durchfluss .....	175
6.4.2	Constraints und Puffer .....	178
6.4.3	Cost of Delay - Projekte priorisieren .....	179

6.5	Das agile Projektmanagement-Office .....	181
6.6	KPIs und Reporting .....	184
6.7	Marketing für das agile Projekt .....	186
<b>7</b>	<b>Die fraktal skalierte Organisation .....</b>	<b>189</b>
7.1	Wer führt die fraktal skalierte Organisation? .....	193
7.2	Ein Leitfaden für die fraktal skalierte Organisation .....	195
7.2.1	Glaubenssätze und Werte .....	196
7.2.2	Fähigkeiten .....	207
7.2.3	Verhalten .....	215
7.2.4	Umwelt – Kunde, Dienstleister und Regularien .....	216
<b>Schlussendlich: Es fängt mit Ihnen an! .....</b>		<b>219</b>
<b>Literatur .....</b>		<b>223</b>
<b>Index .....</b>		<b>227</b>

# Vorwort

Ich habe dieses Buch für Sie geschrieben. Es wird Sie, so glaube ich, dabei unterstützen, jene Stellen in Ihrem Projekt zu finden, die Sie daran hindern, ein großes agiles Projekt erfolgreich durchzuführen. Lassen Sie mich Ihnen erklären, aus welchen Motiven ich dieses Buch geschrieben habe.

2005 hatte mich der Leiter eines großen Softwareprojekts für Mobiltelefone eingeladen. Er wollte dieses Projekt agil managen: 180 Entwickler, Projektmanager und Tester an drei Standorten. Er fragte mich, wie ich das machen würde. Ich war – und er wusste das – blank. Ich hatte keine Ahnung. Bis zu diesem Zeitpunkt hatte ich Projekte mit maximal 40 Entwicklern gemanagt, zwar an zwei Standorten, aber das war zu vernachlässigen. Ich druckste herum und dann provozierte er mich: Ich solle nicht um den heißen Brei herumreden, sondern klar sagen, wie das funktionieren könnte. Also ging ich zum Whiteboard und malte aus dem Stehgreif den ersten Skalierungsblueprint. Wir bekamen den Auftrag. Für einige Monate hatten wir die Chance, mit Menschen zu arbeiten, die noch nie Scrum gemacht hatten, sich gar nicht vorstellen konnten, was Agilität war und noch dazu alte Prozesse einhalten mussten. Und natürlich saßen Tester und Entwickler in verschiedenen Abteilungen. In diesem Projekt entstanden die ersten Ideen zu den Product Owner-Teams, zum kontinuierlichen Testen und später auch die Ideen zu gemeinsamen Retrospektiven. Wir spielten in diesem Projekt zum ersten Mal Planning Poker und entwickelten die ersten großen Release-Pläne. Damals wie heute war das Skalieren im Grunde einfach. Schwierig zu bewältigen waren die Widerstände gegen das agile Arbeiten, auf die wir in der Organisation immer wieder trafen.

Doch mit dieser Aussage lässt sich kein Geld verdienen. Der Scrum-Beratungsmarkt hat sich darauf verlegt, den Kunden Methoden und Produkte zu verkaufen, die zwar gut für die Berater, aber nicht für die Kunden sind. Es ist viel einfacher, Trainings und Zertifizierungen zu verkaufen. Modelle schießen wie Pilze aus dem Boden: LeSS, SAFe®, Nexus™ und wie sie alle heißen. Man trainiert in zwei Tagen ein neues Vokabular und in manchen Unternehmen wird dann auch über diese Modelle gesprochen. Die Modelle sind kompliziert, viel zu kompliziert. Sie sind nichts anderes als komprimierte Best Practices, so wie mein Bild damals auf dem Whiteboard. Allerdings tun diese Modelle eines nicht: Sie skalieren ein Projekt nicht.

Ich wollte da nicht mitmachen, zumal ich weiß, dass es nicht das Framework ist, das ein Projekt erfolgreich macht. Anders als das simple Scrum-Framework, das Anfängern den

Zugang zum agilen Denken eröffnet, sind diese Modelle nicht dazu geeignet, das agile Denken in einem Unternehmen zu fördern. Das hat mich lange daran gehindert, selbst etwas zum Thema Skalierung zu schreiben. Ich wollte nicht noch eine Blaupause liefern, die nicht funktioniert. Außerdem spürte ich zwar, was notwendig ist, um ein großes Projekt vernünftig durchführen zu können, ich konnte das außerhalb der Praxis aber nicht artikulieren. Dieses Wissen war noch implizit. Erst als ich mich dazu zwang, dieses Buch zu schreiben (und tatsächlich bin ich ein halbes Jahr lang immer wieder davor weggelaufen), ging mir irgendwann ein Licht auf. Da war es: ein einfaches pyramidales Modell. Doch es ist kein Framework, sondern vielmehr ein Reifemodell: Was muss man in einem Projekt beherrschen, damit Skalierung tatsächlich funktioniert? Worauf müssen Projektleiter, Manager oder CXOs achten, um sicherzustellen, dass ein großes agiles Projekt funktioniert?

Das Schreiben hat leider noch einmal ein ganzes Jahr gedauert. Viele Aspekte habe ich hin und her gedreht, ich habe überlegt, ob ich es an anderer Stelle schon einmal ähnlich beschrieben habe und überhaupt noch einmal einbringen soll. Ich habe mich dafür entschieden, weil es für die Darstellung des Gesamtbilds notwendig war. Und doch habe ich kein Buch für Anfänger geschrieben. Daher eine Warnung: Wenn Sie noch nie ein Scrum-Buch gelesen haben, wenn Sie noch nie ein agiles Projekt durchgeführt haben, werde ich Sie enttäuschen. Ich habe kein Kochbuch geschrieben. Es ist vielmehr eine Landkarte, auf der ich Wegmarkierungen hinterlassen habe und wichtige Stationen beschreibe – allerdings müssen Sie Ihren eigenen Weg beschreiten, immer die eigene Implementierung suchen. Legen Sie das Buch wieder weg, wenn Sie gerade erst mit Scrum anfangen. Sie werden mich verfluchen und ständig den Kopf schütteln. Sie finden hier keinerlei anwendbare Regeln, die Sie erfolgreich machen, sondern nur Anregungen, die Sie erst verstehen werden, wenn Sie sich vorstellen können, wie ein Scrum-Team arbeitet.

Es gibt eine Ausnahme: Sollten Sie ein CXO sein, dann wird Ihnen dieses Buch dabei helfen, zu verstehen, worauf Sie sich einlassen. Ich hätte es gerne schon vor Jahren meinen Kunden in die Hand gedrückt und gesagt: „Wenn du *das* willst, dann fangen wir an.“ Ich bin mir sicher, die meisten hätten mir nicht geglaubt oder wären so schockiert gewesen, dass sie die lange Reise zur agilen Organisation nie angetreten hätten. Doch alles, was zwischen den Deckeln dieses Buchs steht, wird auf Sie zukommen oder betrifft Ihre Organisation, wenn Sie Scrum wirklich skalieren wollen. Den Reiseführer habe ich geschrieben – losgehen müssen Sie selbst.

Wie meine anderen Bücher wäre auch dieses nicht ohne die vielen Stunden Arbeit von Dolores Omann geschrieben worden. Unermüdlich hinterfragt sie meine Gedankengänge, korrigiert meine Rechtschreibung und macht aus Bandwurmsätzen lesbare Texte. Sie kürzt und sagt mir immer wieder, wo ich noch deutlicher werden musste. Vielen Dank Dolores.

Vielen Dank an meinen besten Freund Andreas de Pretis, Geschäftsführer der 25<sup>th</sup>-floor GmbH, der etwas Technik in das Buch eingebracht hat. Danke auch an Vincent Tietz von Saxonia Systems AG, der mit seinem Beitrag zeigt, welche Veränderungen und Vorwärtsbewegungen für ein Unternehmen möglich werden, wenn das agile Arbeiten im Topmanagement ankommt.

Mein Dank geht aber vor allem an die Kolleginnen und Kollegen in unserem Unternehmen borisgloge consulting. Sie stehen jeden Tag ihre Frau und ihren Mann und sind mit all den Problemen und Widerständen konfrontiert, über die Sie auf den folgenden Seiten lesen werden. Sie meistern diese Herausforderungen jeden Tag ein Stück mehr und wissen, dass der

Weg weiterhin steinig bleiben wird. Ohne sie, ohne ihre Arbeit, wäre dieses Buch nichts weiter als eine nette Geschichte, unüberprüft und nicht real. Dank ihrer Arbeit ist es Realität.

Ich danke unseren Kunden, die sich mit uns auf eine abenteuerliche Reise gemacht haben. Ich danke Ihnen für Ihr Vertrauen in uns und für den Mut, oft mehr darauf zu vertrauen als zu wissen, dass die Reise erfolgreich sein wird. Ich danke Ihnen dafür, dass Sie Ihre Ängste uns gegenüber äußern und wir mit Ihnen gemeinsam immer wieder neue Erfahrungen machen dürfen. Denn ja: Jede Skalierung ist tatsächlich anders.

Ich danke meiner Frau Kathrin, die mich immer unterstützt. Sie sitzt mir gerade gegenüber und akzeptiert, dass ich in diesem Moment unseres Urlaubs nicht mit ihr rede, sondern dieses Vorwort schreibe.

Viel Spaß beim Lesen und viel Erfolg beim Selbst-Skalieren von Scrum!

*Boris Gloger*

Wien, Frühjahr 2017

# Über den Autor



**Boris Gloer** 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 borisgloer consulting GmbH mit Sitz in Baden-Baden gründete, war der Unternehmer als Business Analyst, Team-Leader, Projektmanager und Scrum Consultant für globale Unternehmen (z. B. EDS, Nokia, BenQ) tätig. borisgloer consulting bietet Training und Consulting zur agilen Produkt- und Organisationsentwicklung mit Scrum sowie zum agilen Management für Fach- und Führungskräfte an. Boris Gloer gilt inzwischen als einer der progressivsten Denker im Bereich Management und Organisation und ist ein gefragter Vortragender auf Managementkonferenzen.

Folgende Bücher von Boris Gloer sind im Carl Hanser Verlag erschienen:

- Scrum. Produkte zuverlässig und schnell entwickeln. 5. Aufl. Carl Hanser Verlag, 2016.
- Selbstorganisation braucht Führung. Die einfachen Geheimnisse agilen Managements. Carl Hanser Verlag, 2014.
- Wie schätzt man in agilen Projekten – oder wieso Scrum-Projekte erfolgreicher sind. Carl Hanser Verlag, 2014.
- Der agile Festpreis. Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge. 2. Aufl. Carl Hanser Verlag, 2014.
- Erfolgreich mit Scrum: Einflussfaktor Personalmanagement. Finden und Binden von Mitarbeitern in agilen Unternehmen. Carl Hanser Verlag, 2011.

Kontakt: [boris.gloer@borisgloer.com](mailto:boris.gloer@borisgloer.com)

# Die Umfeldbedingungen des Skalierens

„Funktioniert Scrum auch mit großen Projekten?“, fragen mich die Teilnehmerinnen und Teilnehmer in Seminaren, unsere Kunden und so gut wie jeder Manager, den ich treffe. Sie alle wollen ihre großen, aus dem Ruder gelaufenen Projekte retten. Immer wieder muss ich dabei an den Chaos Report der Standish Group denken (vgl. Standish Group 2015): Mehr als 50 Prozent aller großen Projekte liefern nicht die geforderten Ergebnisse. Die Diagnose ist eindeutig: Je größer ein Projekt ist, desto wahrscheinlicher wird es – gemessen an den ursprünglichen Vorgaben – scheitern. Übrig bleibt das Fazit: Obwohl alle Beteiligten bestens im Projektmanagement ausgebildet sind, obwohl versucht wird, am Beginn eines Projekts an möglichst viel zu denken, steckt das traditionelle Management großer Projekte in einer Sackgasse.

In diesem Buch beschreibe ich einen Ausweg: Wie gelingen große Projekte in der Soft- und Hardwareentwicklung? Als großes Projekt bezeichne ich eines, das folgende Bedingungen erfüllt: Ein Budget von mehr als einer Million Euro, eine Dauer von mehr als sechs Monaten und mehr als sechs durchführende Beteiligte, die auch an verschiedenen Orten angesiedelt sein können. Wie solche Projekte gemanagt werden können, ist eine brennende Frage, denn es ist noch immer die größte Herausforderung für die meisten Unternehmen.

Großkonzerne und der deutsche Mittelstand führen Projekte in gigantischen Ausmaßen durch – zehn Millionen Euro sind schnell ausgegeben und Dutzende Personen leisten ihren Beitrag. Aber da gibt es noch einige Besonderheiten: Diese Projekte werden im Rahmen des normalen Tagesgeschäfts durchgeführt, sie stören den normalen Ablauf. Arbeiten mehrere Standorte am Projekt mit, ist die Projektsprache Englisch. Und als wäre das allein nicht schwierig genug, sind diese Unternehmen noch dazu in einem gigantischen Beschleunigungsrennen gefangen. „Digitalisierung“ lautet das Stichwort, täglich entstehen neue Technologien. Neue Optionen, neue Geschäftsfelder entwickeln sich über Nacht. Unternehmen, die auf dem internationalen Parkett weiter bestehen wollen, müssen damit umgehen können. Fehlentscheidungen und Selbstzufriedenheit führen schnell zum Desaster.

Mit der Flut an Möglichkeiten, die neue Technologien mit sich bringen, rast eine Welle von Gesetzesänderungen auf die Unternehmen zu. Kontinuierlich müssen sie ihre Prozesse und damit oft ihre IT-Infrastruktur anpassen, und natürlich müssen auch Projekte neu bewertet und ihr Kurs unter Umständen korrigiert werden. Vieles bleibt lange im Ungewissen, denn manchmal wissen nicht einmal die Behörden selbst, wie eine neue Vorschrift aussehen soll. Man weiß, dass sie kommen wird, aber nicht, wann genau und in welcher Form. Werden die entwickelten Lösungen die richtigen sein?

In diesem Spiel kann nur mitmischen, wer seine Methoden den Umständen anpasst. Traditionelles Projektmanagement muss zwangsläufig scheitern, denn dessen Spielwiese wurde für einen anderen Kontext und andere Zeithorizonte entwickelt. Es leistete gute Dienste, viele Unternehmen haben gute Erfahrungen damit gemacht. Aber immer wieder ist zu beobachten, dass extrem kritische Projekte außerhalb der traditionellen Unternehmensstrukturen durchgeführt werden. Die Labs und Taskforces sind anderen Bedingungen unterworfen: Dort dürfen die Kollegen alles anders machen und werden nur daran gemessen, ob sie eine Lösung finden. Klassische Unternehmensprozesse werden ignoriert oder so geschickt ausgenutzt, dass sonst Undenkbares plötzlich möglich wird.

Die Softwareentwicklung war der erste Bereich, in dem in den 1990er-Jahren sichtbar wurde: Der traditionelle Weg der Projektdurchführung funktioniert nicht mehr. Also machten sich einige Manager auf den Weg, suchten eine Lösung für die beschriebenen Probleme und fanden die agile Softwareentwicklung. Von den unterschiedlichen Modellen setzte sich weltweit eine Vorgehensweise durch: Scrum. Dieses Managementframework wurde entwickelt, um in einem komplexen Umfeld möglichst schnell etwas liefern zu können, von dem man oft nicht weiß, wie es aussehen soll und erreicht werden kann. Zunächst ging man davon aus, dass man gut ausgebildete Softwarespezialisten zusammenbringen müsse, um aus ihnen ein Team mit einer klar definierten Aufgabe zu formen. Das funktionierte sehr gut. Heute arbeiten viele Unternehmen in ihren IT-Abteilungen erfolgreich mit diesen kleinen, sich selbst organisierenden Teams.

Scrum-Projekte wurden bald so erfolgreich, dass immer mehr Hardwareentwicklungsteams auf diese Arbeitsweise aufmerksam wurden. Die Softwareentwickler lieferten ihre Produkte immer schneller ab, nutzten andere Formen des Reportings und wirkten alles in allem zufriedener mit ihrer Arbeit. So hielt Scrum auch Einzug in das traditionelle Projektmanagement für Hardwareprodukte. Und die Verbreitung von Scrum ist noch nicht abgeschlossen: Inzwischen gibt es selbst verwaltungsnahe agile Teams, zum Beispiel in der Rechtsabteilung der Holtzbrinck Publishing Group (vgl. Gloger 2015).

### **Geht das auch groß?**

Ein kleines, sich selbst überlassenes Team kann bei einer klar umrissenen Aufgabe alle paar Wochen etwas vorzeigen. Das kann man sich vorstellen und das ist auch mit klassischem Projektmanagement erfolgreich möglich. Scrum-Projektteams wurden aber schnell mit der Frage konfrontiert, wie sie denn große Projekte stemmen würden, denn genau das ist die Herausforderung.

Ich habe mich selbst lange immer wieder dagegen gesträubt, diese Frage zu beantworten. Meine Sicht der Dinge war einfach: Kleine Scrum-Teams aus gut ausgebildeten Mitarbeitern sind um einiges effektiver als große behäbige Projektteams. Also, weshalb sollte es den Bedarf geben, Projekte möglichst groß aufzuziehen? Die Antwort lag doch auf der Hand: Man musste einfach die Projektteams verkleinern. Ich hatte selbst gesehen, dass ein zehnköpfiges Team in kurzer Zeit Projekte ablieferte, an denen 150 Personen gescheitert waren. Alles sprach also dafür, dass man das große Projekt gar nicht brauchte.

Mir wurde erst klar, dass ich mich mit dem Thema noch einmal ganz anders auseinandersetzen musste, als die Anfragen zu Scrum in der Hardwareentwicklung immer häufiger wurden. Plötzlich waren völlig andere Probleme zu lösen: In diesem Bereich arbeiten nicht zwangsläufig Menschen zusammen, die ähnlich denken. Manche „verstehen“ sich aufgrund

ihrer Ausbildung überhaupt nicht. Chemiker denken anders als Biologen, die wiederum ganz anders an Probleme herangehen als Elektroingenieure, die natürlich alles ganz anders machen als die Maschinenbauer. Dazu kommt, dass diese unterschiedlichen Disziplinen oft gar nicht an einem Standort angesiedelt sind. Die wunderbare Idee, alle Kollegen in einem Raum zu versammeln, ist oft gar nicht umsetzbar, denn die Spezialisten sind in der ganzen Welt verteilt und mitunter gehören sie nicht einmal zum eigenen Unternehmen. Aber das Internet lässt die Welt auch in diesem Punkt schrumpfen: Die besten Köpfe lassen sich leichter finden als je zuvor und sind oft kurzfristig buchbar. Ein befreundetes kleines Zweimann-Unternehmen mit einem stark eingegrenzten Spezialgebiet arbeitet für einen kalifornischen Giganten. Die zwei nehmen Aufträge an, liefern die Ergebnisse ab und es scheint hervorragend zu funktionieren. Längst haben „New Brokers of Work“ daraus ein Geschäft gemacht: Sie vermitteln Aufgaben an Spezialisten, finden diese in weltweiten Netzwerken oder bauen selbst diese Netzwerke auf. Das ist auch Teil des sich abzeichnenden Trends zur Hyperspezialisierung: Arbeiten in Projekten werden kleinteilig auf diejenigen verteilt, die sich am besten für die Aufgabe eignen (vgl. Malone et al. 2011).

Wer genau hinsieht, macht eine erstaunliche Beobachtung: Wieder ist es die Softwareentwicklung, in der dieser Trend seinen Ausgang nimmt. Weltweit verteilte Softwareentwickler arbeiten gemeinsam an Projekten, obwohl sie sich oft noch nie persönlich gesehen haben und manchmal nicht einmal die Namen der anderen kennen. Unternehmen wie Automattic (Wordpress) oder 37signals (Basecamp) wurden bewusst so konzipiert. Sie wollen mit Menschen auf der ganzen Welt zusammenarbeiten, die das benötigte Wissen haben. Microsoft baut viele seiner neuen Produkte konsequent in die Cloud, damit Menschen in jedem Winkel der Erde daran arbeiten können, auch Adobe setzt auf das gleiche Prinzip.

### **Schwieriger Wandel im Projektmanagement**

Obwohl heute alle vor den gleichen Problemen stehen wie seinerzeit die Softwareentwicklung, gehen viele Unternehmen nach wie vor den aus meiner Sicht traditionellen und damit ungeeigneten Weg:

- Projekte werden aus der Taufe gehoben und erst dann wird überlegt, wer die Aufgaben am besten erledigen kann.
- Man stellt Ressourcenmanager ein, die Projektmanager immer mit den richtigen Ressourcen versorgen sollen.

Doch leider zeigt die Realität: In Projekten werden immer wieder die gleichen Personen benötigt, die genau deswegen bereits in vielen Projekten gleichzeitig arbeiten. Die Gründe sind bekannt:

1. Es werden zu viele Projekte zur selben Zeit durchgeführt (siehe dazu später die Ausführungen zur Theory of Constraints) und
2. die Key Player im Unternehmen sind entscheidende Wissensträger. Bei ihnen, vielmehr „in“ ihnen, konzentriert sich das Wissen und daher werden sie in mehr als einem Projekt gebraucht.

Intuitiv ist das allen Managern klar. Daher war der immer wieder entstehende Widerstand nachvollziehbar, wenn wir forderten, dass in einem Scrum-Projekt alle gemeinsam an *einem* Projekt/Produkt arbeiten sollten. Wie sollte das gehen? Denn gelebte Realität war und ist in vielen Organisationen, dass ein Kollege immer in mehr als einem Projekt arbeiten muss.

Kleine Erfolge waren dennoch möglich. Bei ganz großen Projekten brachte ich das Management oft dazu, alle Projektbeteiligten zusammenzurufen. Das eine oder andere Mal führten wir Sprint Plannings mit 180 Personen in Ballsälen durch oder wir konnten Entwickler, die auf einem Stockwerk verteilt saßen, durch geeignete Maßnahmen zu einer besseren Zusammenarbeit motivieren. Das waren Ausnahmen, die zwar bewiesen, dass viele meiner Ideen korrekt waren und vor allem schlängernde Projekte konnten wir so wieder auf Kurs bringen. In der Masse waren diese Projekte aber nur ein Tropfen auf den heißen Stein. Sie funktionierten und brachten großartige Erfolge, aber die Herangehensweise war für die meisten Unternehmen nicht wiederholbar. Für dieses eine Projekt war die Kraftanstrengung möglich, die ich forderte, in letzter Konsequenz hätte es aber eine Änderung der Organisation bedeutet. Viele Unternehmen verstehen das erst, nachdem sie die Erfolge von Scrum-Projekten gesehen haben.

Und doch geschieht es: Die großen Tanker ändern ihren Kurs. Die vielen kleinen erfolgreichen Scrum-Implementierungen am Rande der Organisationen haben den Verantwortlichen gezeigt, dass sich grundsätzlich etwas ändern muss, wenn ein Unternehmen erfolgreich bleiben soll. Sie stellen erste Überlegungen dazu an, wie die Struktur der gesamten Organisation umgestaltet werden muss (vgl. Gloger, Margetich 2014).

Änderungen der Organisation können gelingen und machen das Projektleben leichter, doch das ist heute noch nicht die Regel. Die meisten großen Projekte werden in nichtagilen Umgebungen durchgeführt. Ist damit die Chance auf ein agiles Projekt verspielt? Nein, es ist möglich, agile Projekte in einem nichtagilen Umfeld über die Ziellinie zu bringen. Dazu muss Zusammenarbeit neu definiert werden und es bedeutet, bisher bewährte, aber nun veraltete Paradigmen allmählich aufzugeben.

## Ein erster Überblick

Dem großen Projekt müssen wir uns also noch einmal anders nähern. Nicht mit den naiven Ideen der Scrum-Evangelisten, zu denen auch ich gehört habe, sondern mit dem gesamten agilen Erfahrungsschatz der letzten 15 Jahre. Beim Skalieren eines großen Projekts geht es in Wahrheit immer um die Entwicklung von etwas Bleibendem. Das kann ein neuer Prozess, ein neues Gebäude, ein neues Software- oder Hardwareprodukt sein. Jedes Mal wird am Ende ein Resultat erwartet. Es reicht also nicht aus, sich lediglich mit dem Scrum-Prozess zu beschäftigen. Wir müssen ein größeres Bild zeichnen und uns mit den folgenden Themen auseinandersetzen:

**Wissensmanagement.** Ikujiro Nonaka und Hirotaka Takeuchi sowie Peter Drucker haben für das Verständnis der Wissensarbeit Großartiges geleistet. Ihre Forschung und ihre Überlegungen machten deutlich, warum gerade die Softwareentwicklung neue Impulse setzen konnte und musste.

**Softwarearchitektur.** Ausgehend vom Silicon Valley verbreitet sich das Denken in Netzwerken und kleinen Einheiten, und das begründet wiederum ein neues Verständnis für die Zusammenarbeit von Menschen.

**Lean Management, Lean Production und Second Generation Lean Product Development.** Wer große Projekte agil umsetzen will, kommt nicht umhin, sich ausführlich mit dem Toyota Production System, der Theory of Constraints und den Arbeiten von Don Reinertsen zu beschäftigen. Auf Grundlage dieser Ideen wird deutlich, warum der Wertstrom und nicht der Plan die relevante Kenngröße für das Steuern eines Projekts ist.

**Design Thinking.** Die Revolution der Produktentwicklung hat in den 1980er-Jahren in Kalifornien begonnen. Die Designagentur IDEO entwickelte damals ein Vorgehensmodell, das tatsächlich die Anwender (User) in den Mittelpunkt aller Überlegungen stellt. IDEO verabschiedete sich von dem Konzept, dass eine Person die Produktidee haben muss und die Umsetzung daher vorgibt. Design beschreibt in diesem Sinne nicht das Erscheinungsbild eines Produkts, sondern das Verständnis, wie ein Anwender ein Produkt in seinem Lebenskontext nutzt.

**Führung.** In großen Projekten sind die beteiligten Menschen oft überall auf der Erde verstreut und man bekommt sie so gut wie nie zu Gesicht. Die Frage, wie man Menschen führen soll, bekommt hier eine völlig neue Dimension. Wir werden sehen, dass klare Kommunikation und ausschließlich intrinsische Faktoren den Ausschlag geben.

**Organisation.** Wie verändert sich die Organisation rund um ein großes agiles Projekt? Natürlich kann nicht gleich ein ganzes Unternehmen plötzlich agil werden und trotzdem greifen solche Projekte die Abteilungssilos an.

Ich bin der Meinung: Mit agilem Projektmanagement für das große Projekt kann man sich nur befassen, wenn man die folgenden sechs Bausteine anders als bisher versteht:

- Produktarchitektur
- Infrastruktur
- Professionalität
- Produktentwicklungsprozesse
- Managementprozesse
- Den Begriff des Projekts

Diese sechs Bereiche werde ich in den kommenden Kapiteln genauer ausleuchten. Stecken wir aber zunächst den Rahmen ab. Sehen wir uns an, unter welchen Voraussetzungen heute Skalierung, also das verteilte, multidisziplinäre miteinander Arbeiten in großen Projekten passiert.

## ■ 1.1 Hyperspezialisierung

Es ist der Ursprung vieler Probleme: die Spezialisierung bzw. das Arbeiten in Silos. Alle achten auf ihre eigenen kleinen Aufgabenbereiche, wichtig ist nur, ob es in der eigenen Abteilung richtig läuft. In den letzten Jahren hat sich diese Problematik noch verschärft, weil wir es inzwischen mit „Hyperspezialisierung“ zu tun haben. Die einzelnen Wissensdomains und Verantwortungsbereiche sind oft schon so klein, dass immer weniger Menschen immer mehr wissen.

Die Idee der Arbeitsteilung war in den Anfängen der Industrialisierung sehr nützlich. Sie hat zu immensen Produktivitätssteigerungen geführt und sie erfüllt nach wie vor in der Fertigung von Gütern ihren Zweck. Wie Malone, Laubacher und Johns im Artikel „The Big Idea: The Age of Hyperspecialisation“ schreiben, ist aber auch die Wissensarbeit zu einem Feld der Arbeitsteilung geworden und sie passiert hier in einem noch viel größeren Ausmaß

als in der Produktion (vgl. Malone et al. 2011). Für den Boeing 787 Dreamliner gab es alleine 379 Hauptlieferanten und dementsprechend schwierig und fehlerreich war der Zusammenbau. Doch in diesem Fall kann man wenigstens sehen, wo das Problem liegt. Wissensarbeit bietet noch viel mehr Möglichkeiten, Arbeit in unzählige Stücke zu schneiden, denn die Arbeit ist nicht greifbar und kann mühelos und kostengünstig durch die ganze Welt transportiert werden. So wie Handwerksberufe in ihre Einzelschritte zerlegt wurden, wird jetzt die Arbeit der Ingenieure, Softwareentwickler, Marketingverantwortlichen etc. atomisiert und in ein weltweites Netz von Spezialisten verwandelt.

Der Hyperspezialisierung, wie Malone, Jones und Laubacher sie beschreiben, steht allerdings der Trend zum Generalisten gegenüber. Die Vertreter des Design Thinkings setzen auf „T-Shaped People“, also auf Mitarbeiter mit einem allgemeinen Wissen in mehreren Bereichen und tiefgreifendem Wissen in einem Spezialgebiet. In diesem Spannungsfeld bewegen sich die Teilnehmer in internationalen Projekten: Einerseits wird immer der beste Experte für eine Spezialaufgabe gebraucht und andererseits braucht man die Wendigkeit von Generalisten für Probleme, die sich von Tag zu Tag ändern. Immer den richtigen Spezialisten an Bord zu haben, ist unmöglich und trotzdem muss man wissen, dass und wann man einen Spezialisten braucht.

**Fragmentierung.** Wie Malone, Laubacher und Johns richtig anführen, ist die Fragmentierung, also die Spezialisierung in der Spezialisierung, inzwischen sehr hoch. An der Softwareentwicklung wird das sehr deutlich, denn innerhalb dieser Disziplin gibt es wiederum Spezialisierungen auf fachlicher Ebene (Programmiersprachen, Architekturkonzepte, Frameworks etc.), in den Prozessschritten (Analyse, Design, Implementierung, Testing, Betrieb) und in den Branchen. Dasselbe gilt auch für andere Disziplinen wie den Maschinenbau, den Fahrzeugbau, die Luft- und Raumfahrttechnik usw.

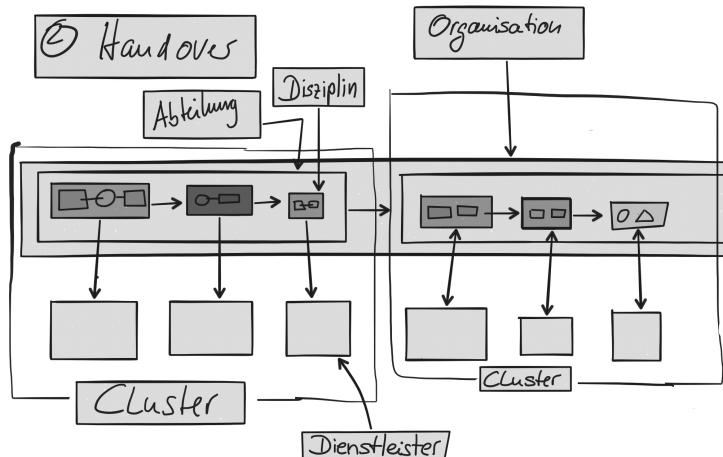
**Intraorganisationale Hyperspezialisierung.** Was in den Disziplinen anfängt, setzt sich auch in den Organisationen fort. Unternehmen strukturieren sich nach verschiedenen Disziplinen – HR, Entwicklung, Marketing etc. Innerhalb dieser Abteilungsdisziplinen entstehen durch die Prozesse wiederum Spezialisierungen. In großen Unternehmen ist das besonders deutlich zu beobachten: Fachabteilungen diskutieren ihre Ideen mit Organisationsabteilungen wie etwa der IT, deren Aufgabe es ist, die Anforderungen aufzunehmen. Diese bespricht die Anforderungen wiederum mit den Entwicklungsabteilungen, die ihrerseits wiederum in Prozessschritte unterteilt sind. So gibt es in der Softwareentwicklung Abteilungen, die sich nur um Anforderungen und die Planung kümmern, die nur entwickeln oder die nur testen.

**Interorganisationale Hyperspezialisierung.** Und schließlich wird Hyperspezialisierung auch im großen Verbund deutlich. Große Organisationen und ihre Dienstleister arbeiten oft in Clustern. Interne Abteilungen sind dann angehalten, ausschließlich mit Unternehmen zu arbeiten, die Teil des Clusters sind.

Die Idee hinter allen Erscheinungsformen der Hyperspezialisierung ist einleuchtend: Jeder macht, was er am besten kann, und garantiert damit das beste Ergebnis. Aber die Sache hat einen Haken: Hyperspezialisierung lässt viele Übergabepunkte entstehen, sogenannte Handover. Diese Punkte erfordern auf beiden Seiten, dass sich eine Schnittstellenfunktionalität – meist in der Person eines Projektmanagers – um das Verwalten und Sicherstellen einer korrekten Informationsübergabe kümmert. Projekte, die bereits durch ihre Rahmenbedingungen (z. B. Technologie und Regularien) komplex sind, werden dadurch auch organisatorisch extrem komplex. Mitarbeiter nehmen diese Komplexität auf mehrere Arten wahr:

- Hyperspezialisierung führt zu einer **Meetinginflation**. Jeder Handover lässt einen Kommunikationsoverhead entstehen, denn die Beteiligten müssen sich ständig abstimmen.
  - Das führt zu dem unangenehmen latenten Gefühl, möglicherweise jemanden vergessen zu haben.
  - Dieses Gefühl wiederum führt zu einer Unsicherheit, die man durch das Einbeziehen möglichst vieler Beteiligter zu reduzieren versucht.
  - Weil nun so viele Personen an den Meetings teilnehmen, muss das Besprochene genau dokumentiert werden, damit man sich auch noch in einigen Monaten daran erinnert.
- Die Spezialisierung des Wissens führt dazu, dass der Einzelne immer weniger über den **Gesamtzusammenhang des Projekts** weiß.
  - Der Überblick geht verloren.
  - Jeder Mitarbeiter, jeder Manager empfindet sich am Ende nur noch als kleines Rädchen in einer gigantischen Maschine.
  - Als Konsequenz wird immer unklarer, warum man morgens für dieses Projekt eigentlich aufsteht - der Sinn geht verloren.
- Die Masse an Menschen in diesen Projekten erzeugt ein Gefühl der **Heimatlosigkeit**.
  - Zu Menschen, mit denen man im Grunde nichts zu tun hat, entwickelt man keine echte Beziehung.
  - Das Gefühl der Gemeinschaft geht verloren.
  - Die für Unternehmen spürbarste Konsequenz: **Projekte werden langsamer abgewickelt**. Dadurch verringert sich der Return on Investment. Mitunter gelingt es sogar, Projekte trotz stetiger Planänderungen abzuwickeln. Der Projektplan zeigt dann nicht mehr, dass die eigentliche Planung unzureichend war. Es gibt kaum noch Baseline Planning und Projekte werden oft schon gar nicht mehr dagegen gemessen. Also werden die Durchlaufzeiten von Projekten länger und länger. Die Folgen:
    - Projektmitarbeiter identifizieren sich nicht mit dem Projekt, denn mit hoher Wahrscheinlichkeit werden sie das Ende des Projekts nicht erleben - voraussichtlich befinden sie sich dann bereits in einem anderen Projekt.
    - Projekte werden teurer, da sie zwischendurch gestoppt werden müssen. Die nötigen Ressourcen werden auch in anderen Projekten gebraucht. Die Kosten, die durch die Wiederaufnahme des Projekts entstehen, werden oft nicht gesehen oder sogar negiert.
    - Jede Verzögerung kostet Geld. Aus einem ganz einfachen Grund ist die Reduktion der Durchlaufzeiten oberstes Prinzip im Lean Management: Es bringt sehr viel Geld. Je schneller ein Stück geliefert werden kann, desto schneller kann damit Geld verdient werden. Das erhöht den Return on Investment.

Auf den Punkt gebracht: Ständige Handover stören den Arbeitsfluss eines Projekts (Bild 1.1). Sie führen zu Wartezeiten, Missverständnissen und politischen Spielchen.



**Bild 1.1** Handover bremsen das System aus

Das Traurige ist: Menschen *wollen* effektiv arbeiten. Sie mögen es nicht, auf andere warten zu müssen. Sie wollen nicht, dass ihnen andere etwas vorschreiben und sie ihre eigenen Ideen nicht einbringen können. Daher wird es jeder Einzelne in diesem System vermeiden, einen Handover zu bedienen. Es ist viel einfacher, lange vor sich hin zu arbeiten und so zu tun, als müsse man sich erst am Ende abstimmen. So versucht jeder in diesem Geflecht aus Abhängigkeiten, seine eigene Arbeitszeit und seine eigene Arbeit zu optimieren. Das führt zu einer Optimierung der Subsysteme: Jedes für sich ist ideal ausgestattet, aber das Ganze funktioniert nicht effektiv. Schlussendlich verlängert das insgesamt die Durchlaufzeit.

Ziel des agilen Arbeitens ist es, echte Resultate – also schnell – zu liefern. Das bedeutet, zugunsten des Gesamtsystems in Kauf zu nehmen, dass einzelne Subsysteme möglicherweise nicht optimal arbeiten können. Der Durchfluss durch das System steht an oberster Stelle.



### Lean Management = agiles Management?

Wenn Sie die Entwicklung der Agile Community mitverfolgt haben, haben Sie sicher bemerkt, dass einige Autoren Lean Management und agiles Management gemeinsam denken und manchmal sogar synonym verwenden. Ich gehöre nicht zu diesen Autoren. Zwar spreche ich oft von Verfahren, die aus dem Lean Management bekannt sind, aber damit will ich nicht sagen, dass Lean und Agile dasselbe sind. Vielmehr kann das Lean Management vor allem auf prozessueller Ebene viele Wirkweisen des agilen Managements erklären.

Agile Verfahren sind aber nicht identisch mit Verfahren des Lean Managements. Man kann Lösungen des Lean Managements nicht unverändert auf den agilen Kontext übertragen, weil Lean zum Beispiel die Varianz und Variabilität aus einem Produktionssystem entfernen will. Alle Prozesse werden so lange optimiert, bis sie überall identisch sind. Es werden möglichst wenige Varianten entwickelt. So befinden sich zum Beispiel in vielen Autos die gleichen Motoren, lediglich die Steuerungssoftware sorgt für einen Unterschied.

In der Produktentwicklung stellt sich jedoch die Kraft des Faktischen dagegen: Anders als in Produktionsprozessen lässt sich hier weder die Varianz noch die Variabilität reduzieren. Weder der Zeitpunkt der Geburt einer neuen Idee noch das Ausmaß dieser Idee ist jedes Mal gleich, also stabil. Die Prozesse in der Entwicklung von Produkten müssen nicht nur auf ständige Änderungen reagieren können, sie müssen sogar damit arbeiten! Das agile Management hat für diesen Umstand Antworten entwickelt, indem es unter anderem mit einem Element arbeitet, das es in der klassischen Produktion nicht gibt: Kreativität. Kreativität ist die Kraft des Menschen, unter gegebenen Bedingungen, die auf den ersten Blick nur eine Lösung zulassen, einen weiteren Weg zu finden, der eine bessere Lösung bietet. Wir werden sehen, dass es genau diese Eigenschaft des agilen Arbeitens ist, die emergente Lösungen ermöglicht.

## ■ 1.2 Digitalisierung

Wir nehmen das Smartphone im Schnitt alle 15 Minuten in die Hand und verbringen damit jeden Tag bis zu drei Stunden online. Automobilhersteller integrieren Hotspots in ihre Fahrzeuge und Tesla schickt Routineupdates an seine schnittigen Elektromobile, als wären es gewöhnliche Desktop-PCs. Während wir auf dem Weg nach Hause übers Smartphone schon mal die Heizung in unserer Wohnung aufdrehen, denken Unternehmen darüber nach, wie die Digitalisierung die Wertschöpfungskette verändern wird.

Die Digitalisierung führt zu völlig neuen Herausforderungen: Plötzlich wird jedes Unternehmen zum Softwareunternehmen (vgl. Kirkpatrick 2011). Das hat weitreichende Implikationen: Software und die Infrastruktur, auf der diese Software weltweit verteilt betrieben wird, wird zum essenziellen Element der Wertschöpfungskette. CTOs gewinnen an Bedeutung. Die weltweit wertvollsten Unternehmen werden von Menschen geleitet, die Technik wirklich verstehen. Angegriffen werden die Giganten der deutschen Wirtschaft derzeit von Unternehmen wie Tesla und Salesforce. Im Bereich Consumer Electronic hat der deutsche Mittelstand längst verloren. Noch dominiert Deutschland den Anlagenbau und Industrien, in denen Elektrotechnik bisher wichtiger war als Software (sehr gut zu sehen in der Medizintechnik), doch das wird sich ändern. Software durchzieht alles.

Softwareunternehmen zeigen aber nicht nur vor, wie die Digitalisierung von Produkten funktioniert, sie bestimmen auch die neue Art des Arbeitens. Die internen Arbeitsbeziehungen bei Unternehmen wie Cisco, Evernote, Atlassian oder GitHub sind völlig neu ausgerichtet. Ein Mitarbeiter von Cisco erzählte mir: „In unserem Unternehmen muss jeder zu jedem Zeitpunkt von jedem Ort der Welt aus arbeiten können.“ 2011 veröffentlichte Microsoft das „Manifest für neues Arbeiten“. Es zeigt sehr deutlich, wie ein Unternehmen trotz aller Widerstände des Gesetzgebers die Digitalisierung nach innen verfolgt.

# Index

## A

Abhängigkeiten 64, 89, 157  
Achtsamkeit 113, 122  
Agile Manifesto 18, 196  
Aktionsforschung 20  
Änderungsrate 18  
Anforderungen 125  
Anforderungsmanagement 126  
– agiles 128  
Anwender 127, 130, 135, 141  
Appreciative Inquiry 20  
Arbeitsrecht 96  
Arbeitsschutzgesetze 13  
Architektur 37, 117  
– emergente 41  
Artefakte 167  
Auslastung 108, 179

## B

Backlog 44, 185, 189  
Batchsize 174  
Beschleunigung 42  
Betrieb 61  
Burn-out 14  
Bürokratie 26  
Business-Analyse 126  
Business Model Canvas 176  
Business Value Estimation Meeting  
*siehe* Product Owner Weekly

## C

Change-Team 182  
Chaträume 73  
Checklisten 154  
Coaching 112  
Cone of Uncertainty 24  
Configuration Management 83  
Constraints 44, 49, 108, 140, 142, 174, 178  
Container 83  
Continuous Delivery 82  
Conway's Law 38, 50  
Cost of Delay 148, 151, 177, 179  
Crossfunktionalität 52, 163

## D

Daily Scrum 110, 153  
– skaliertes *siehe* Scrum of Scrums  
Defect Rate 185  
delegieren 120  
Deployment 85  
Deployment-Infrastruktur 82  
Design Thinking 5, 22, 127, 130  
Deskalierung 134, 139  
Developer Experience 83  
DevOps 90f., 135  
DevOps-Kultur 82  
Dev/prod parity 84  
Dialog 59  
Digitalisierung 9  
Discovery 131  
Disziplin 101, 153  
Durchfluss 8, 50, 129, 175, 179  
Durchlaufzeit 170, 179  
Durchsatz 108, 151

**E**

Einkauf 90  
 Einladung 163, 210  
 Engpass 108, 149  
 Entkopplung 44  
 Entscheidungsbefugnis 113  
 Entscheidungsfähigkeit 113, 115  
 Entwicklungsinfrastruktur 82  
 Entwicklungspraktiken  
   - agile 110  
 Entwicklungsteam 128, 143  
   - Skills 116  
 Entwicklungstools 81  
 Estimation Meeting 154  
   - skaliert 161  
 Experiment 205  
 Exploration 133  
 Extreme Manufacturing 12  
 Extreme Programming 66, 109

**F**

Facilitation 153  
 Fähigkeiten 207  
   - soziale 101  
 Features 148  
 Feature-Team 164  
 Feedback 153, 205  
 FIFO 180  
 Flexibilität 13  
 Flipcharts 72  
 Fokus 108  
 Fokusgruppen 182, 184  
 Freiwilligkeit 153  
   - Prinzip der 163  
 Fremdarbeitskräfteregelung 96  
 Führung 164  
   - disziplinarische 142  
   - laterale 143  
 Funktionalität 48, 162

**G**

Generalisten 6  
 Generation Y 11  
 Gesamtprofitabilität 170  
 Gesetz der zwei Füße 164, 212  
 Gesundheit 105

Gilden 59, 74, 162f.  
 Good Practices 139  
 Gruppendruck 14

**H**

Handover 6, 89, 93, 168  
 Holacracy 12  
 Host  
   - einer Gilde 163  
 Hyperspezialisierung 3, 5  
   - interorganisationale 6  
   - intraorganisationale 6

**I**

Ideation 131  
 Impediment 108, 142, 157  
 Informationsstruktur 39  
 Infrastruktur 63  
   - bauliche 68  
 Integration 63  
 Interaktion 77  
 Interesse  
   - intrinsisches 194

**J**

Journal 112

**K**

Kadenz 153  
 Kollaborationstools 75  
 Kommunikationsfähigkeit 101  
 Kommunikationsstruktur 38  
 Kommunikationstools 73  
 Komplexität 59  
 Kontrolle 26f., 172f., 181  
 KPI 184  
 Kreativitätstechniken 132  
 kritischer Pfad 149  
 Kultur des Gelingens 122, 207  
 Kundenzufriedenheit 185

## L

Landeskulturen 122  
Lean Management 4, 8, 108, 174  
Lean Manufacturing 175  
Lean Product Development 175  
Lean Production 4  
Leistungsfähigkeit  
- eines Systems 108, 111  
Leitplanken 140  
Lerntheorie 99  
LeSS 118, 139  
Licht 70  
Lieferanten 39, 60  
- Zusammenarbeit mit 88  
Liefergeschwindigkeit 170

## M

Maker-Szene 87  
Management 16  
- agiles 8, 122  
- mittleres 119  
- Skills 119  
Marktchancen 24  
Material 70  
Medizintechnik 95  
Meetings 152  
Meetinginflation 7  
Meetingräume 71  
Messbarkeit 29  
Microservices 48  
Mikromanagement 172  
Misstrauen 172  
Mitarbeiterzufriedenheit 186  
Mob Programming 31, 56, 102, 116  
Moderation 156  
Multitasking 102  
Mushroom-Architektur 38  
Mut 111, 113

## N

Neinsagen 115, 162  
Net-Promoter-Score 185  
Netzwerkorganisation 32  
Neurolinguistische Programmierung  
195  
NexusTM 139

Nichtwissen 103  
Normen 49

## O

Offenheit 206  
One Piece Flow 108, 156, 174  
Open Space Technology 207  
Organisation  
- agile 182  
- fraktal skalierte 121  
Organisationsarchitektur  
- agile 51  
Output 170

## P

Pairing 56  
Pair Programming 116  
Parallelstrukturen 19  
Performance-Indikatoren 185  
Pinnwände 70  
Planung 205  
Plattform 48  
Policies 92  
Portfoliomanagement 169  
- agiles 139, 169  
- klassisches 171  
Practice Groups 163  
Priorisierung 176, 179  
Product Backlog 125  
Product Owner 109, 128, 141, 147, 152  
- Skills 113  
Product Owner Daily 159  
Product-Owner-Team 114, 162  
Product Owner Weekly 160  
Produktarchitektur 162  
- agile 47  
Produktentwicklung 21, 125  
- neue Formen der 108  
Produktentwicklungsprozess  
- agiler 134  
Produktentwicklungszyklus 17  
Produktidee 127  
Produktindikatoren 185  
Produktinkrement 125  
Produktivität 28, 102 ff., 108, 142, 170  
Produktivitätssteigerung 40  
Produktvision 115

Professionalität 16 f., 97  
 Programmleitung 141  
 Programmvision 142  
 Projekt  
 – verteiltes 81  
 Projektmanagement 3, 17  
 Projektmanagement-Office 141, 165, 176  
 – agiles 181  
 Projektmanager 142, 152  
 Projektplanung 24  
 Projektteam 143  
 Prototyp 127, 133  
 Puffer 178 *siehe* Slack  
 Pull-Prinzip 76

## Q

Qualität 110  
 Qualitätssicherung 41

## R

Rahmenbedingungen 140, 162  
 Raumsituation 68  
 Redundanz 34, 41, 45, 50  
 Refactoring 49, 173  
 Regenschirmeffekt 121  
 Regularien 89, 94, 110, 141  
 Reporting 75, 142, 184  
 Respekt 118  
 Responsive Manifesto 198  
 Retrospektive 80, 112, 154  
 Return on Investment 7, 108, 147, 151, 175  
 Review 154, 185  
 Richtlinien 92  
 Rollen 120, 161  
 Rollenverständnis 109

## S

SAFe® 118, 139  
 Saxonia Systems AG 76  
 Schnittstellen 49  
 Scrum 2, 26, 30, 109, 182  
 Scrum-Board 156  
 ScrumMaster 104, 142, 182  
 – Skills 111  
 ScrumMaster Daily 159  
 ScrumMaster-Team 111, 161

ScrumMaster Weekly 160  
 Scrum of Scrums 154  
 Selbsteinschätzung 102  
 Selbstführung 113  
 Selbstmanagement 101  
 – Techniken 104  
 Selbstorganisation 101, 162  
 Selbstreferenz 14  
 Selbstreflexion 112  
 Silos 5  
 Skalierung  
 – fraktale 190  
 Skalierungsframework 118  
 Skalierungstoolbox 152  
 Skills 15, 21, 97  
 – des Einzelnen 100  
 – des Individuums 15  
 – der Organisation 16  
 Slack 150, 174, 179  
 Small World Network 33, 45  
 SoS *siehe* Scrum of Scrums  
 Spezialisten 6, 21  
 Sprint Planning 1 153  
 Sprint Planning 2 153  
 Stage-Gate-Prozess 22  
 Standards 49  
 Steering Committee 141  
 Steuerung  
 – zentrale 140  
 Story Map 160  
 Strategieentwicklung  
 – agile 77  
 Struktur  
 – formale 213  
 Support-Teams 164  
 Swarming 31  
 Synchronisation 76  
 System  
 – fraktale 121, 190

## T

Taskboard 14, 119  
 Teams 161  
 Teamfähigkeit 101  
 Teamzusammensetzung 59  
 Technik 70  
 Technologie 60  
 Testautomatisierung 67

Test Driven Development 66, 109  
– in der Hardwareentwicklung 67  
Testumgebungen 85  
Theory of Constraints 3, 45, 108, 119, 149, 201  
Timebox 153  
TOC *siehe* Theory of Constraints  
Toyota Production System 108  
Transaktionskosten 134, 191  
Transition-Team 161  
Transparenz 76, 78, 206  
T-Shaped People 6  
Two-Pizza-Team 48

## U

Überkapazitäten 50  
User Story 125, 160

## V

Variabilität 175  
Varianz 175  
Veränderung 41, 182  
Verantwortung  
– der Teams 147  
Verhalten 215  
Verhandlungskompetenz 115  
Vertrauen 144, 191, 204  
Verzögerungskosten *siehe* Cost of Delay  
Videokonferenz 73, 77

Virtuelle Maschinen 83  
Vision 112, 146, 159  
– persönliche 104  
Visualisierung 77

## W

Warteschlangentheorie *siehe* Theory of Constraints  
Wasserfall 25  
Wasserfallmodell 65, 108  
Waste 174  
Weighted Shortest Job First 180  
Werte 174 f.  
– von Scrum 80  
Wertesystem 192  
Wertschöpfungskette 9, 135  
Whiteboard 73  
Widerstand 100  
WIP-Limit 185  
Wissensmanagement 4  
Wissensträger 3  
Work in Progress 174  
– Limit 110

## Z

Zusammenarbeit 75  
– verteilte 77