

# HANSER



## Leseprobe

zu

## Scrum Think big

von Boris Gloger und Carsten Rasche

Print-ISBN: 978-3-446-47612-7

E-Book-ISBN: 978-3-446-47718-6

E-Pub-ISBN: 978-3-446-48089-6

Weitere Informationen und Bestellungen unter

<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446476127>

sowie im Buchhandel

© Carl Hanser Verlag, München

# Inhalt

<b>Vorwort zur 2. Auflage</b> .....	<b>IX</b>
<b>Vorwort zur 1. Auflage</b> .....	<b>XI</b>
<b>Die Autoren</b> .....	<b>XIV</b>
<b>1 Die Umfeldbedingungen des Skalierens</b> .....	<b>1</b>
1.1 Hyperspezialisierung .....	6
1.2 Digitalisierung.....	10
1.3 Arbeitsschutzgesetze gegen das neue Arbeiten.....	14
1.4 Professionalität und neue Skills .....	16
1.5 Produktentwicklungszyklen und Projektmanagement.....	19
1.6 Bürokratie und Kontrolle .....	27
1.7 Ein Lösungshinweis: Die Netzwerkorganisation .....	33
1.8 Kapitelausblick .....	35
<b>2 Organisationsstruktur und Produktarchitektur</b> .....	<b>39</b>
2.1 Architektur als Ergebnis der Kommunikationsstruktur.....	40
2.1.1 Schneller ist besser .....	44
2.1.2 Entkopplung .....	47
2.2 Wie entsteht eine moderne, agile Produktarchitektur? .....	50
2.2.1 Microservices – die neue Grundlage flexibler Produktarchitekturen .....	51
2.2.2 Redundanz und Durchfluss gewährleisten .....	53
2.2.3 Die Einheitlichkeit der Produktarchitektur .....	60
2.2.4 Technologien und Architektur.....	62
2.2.5 Betrieb und Architektur.....	64
<b>3 Infrastruktur</b> .....	<b>67</b>
3.1 Integration ist alles.....	67
3.2 Räume – bauliche Infrastruktur .....	72
3.2.1 Das große Projekt in einem Raum .....	73
3.2.2 Flipcharts, Stifte und Haftnotizen.....	76
3.3 Kommunikationstools.....	77

3.4	Entwicklungstools	85
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	101
4.1.1	Selbstmanagement	102
4.1.2	Wissen über die Theory of Constraints	108
4.1.3	Wissen über neue Formen der Produktentwicklung	109
4.1.4	Die Rollen mit den richtigen Fähigkeiten ausfüllen	112
4.1.4.1	Die Skills des Scrum Masters	112
4.1.4.2	Die Skills des Product Owners	116
4.2	Die Skills des Entwicklungsteams	119
4.3	Die Skills des Managements	121
<b>5</b>	<b>Produktentwicklung</b>	<b>129</b>
5.1	Agiles Anforderungsmanagement	135
5.2	Design Thinking	139
5.3	Lean Startup	145
5.4	Ein agiler Produktentwicklungsprozess	149
<b>6</b>	<b>Managementframeworks für die skalierte Produktentwicklung</b>	<b>153</b>
6.1	Mythos zentrale Steuerung	154
6.1.1	Rahmenbedingungen oder Leitplanken?	155
6.1.2	Steering Committee, Projektmanagement-Office und Projektteams	156
6.1.3	Die Steuerungslogik der Produktorganisation	158
6.2	Das richtige Produkt richtig liefern	160
6.2.1	Team of Teams: gemeinsam für den Return on Investment verantwortlich	163
6.2.2	Theory of Constraints	165
6.3	Die Skalierungstoolbox	168
6.3.1	Die Meetings	169
6.3.1.1	Das skalierte Daily Scrum oder Scrum of Scrums	171
6.3.1.2	Das Scrum Master Daily	175
6.3.1.3	Das Product Owner Daily	175
6.3.1.4	Das Scrum Master Weekly	176
6.3.1.5	Das Product Owner Weekly	177
6.3.1.6	Das skalierte Estimation Meeting	179
6.3.2	Rollen und Teams	180
6.3.2.1	Das Scrum-Master-Team	180
6.3.2.2	Das Product-Owner-Team	181
6.3.2.3	Die Gilden – teamübergreifender Umgang	181
6.3.2.4	Support-Teams	183
6.3.2.5	Das Feature-Team ist der Product Owner	184

6.3.3	Artefakte .....	186
6.3.4	Agiles Portfoliomanagement .....	188
6.3.5	Wert und Durchfluss .....	194
6.3.6	Constraints und Puffer .....	198
6.4	Das agile Projektmanagement-Office .....	200
6.5	KPIs und Reporting .....	203
6.6	Marketing für das agile Projekt .....	205
<b>7</b>	<b>Durch Führung und Werte zur fraktal skalierten Organisation ...</b>	<b>207</b>
7.1	Wer führt die fraktal skalierte Organisation? .....	212
7.2	Ein Leitfaden für die fraktal skalierte Organisation .....	214
7.2.1	Glaubenssätze und Werte .....	215
7.2.2	Fähigkeiten .....	225
7.2.3	Verhalten .....	233
7.2.4	Umwelt – Kunde, Dienstleister und Regularien .....	234
<b>8</b>	<b>Der Weg zur fraktal skalierten Organisation .....</b>	<b>237</b>
8.1	Das skalierte agile Projekt .....	239
8.1.1	Designprinzipien für das skalierte agile Projekt .....	244
8.1.2	Grenzen des skalierten agilen Projekts .....	246
8.2	Die virtuelle Produktorganisation .....	247
8.2.1	Designprinzipien für die virtuelle Produktorganisation .....	255
8.2.2	Grenzen einer virtuellen Produktorganisation .....	257
8.3	Agile End-2-End-Bereiche .....	258
8.3.1	Designprinzipien für agile End-2-End-Bereiche .....	266
8.3.2	Grenzen einer Organisation mit agilen End-2-End-Bereichen .....	268
8.4	Netzwerkorganisation & Microenterprises .....	270
8.4.1	Designprinzipien für Netzwerkorganisationen & Microenterprises .....	279
8.4.2	Grenzen von Netzwerkorganisationen & Microenterprises .....	280
	<b>Schlussendlich: Es fängt mit Ihnen an! .....</b>	<b>283</b>
	<b>Literatur .....</b>	<b>287</b>
	<b>Stichwortverzeichnis .....</b>	<b>293</b>



# Vorwort zur 2. Auflage

Im Vorwort zur ersten Auflage dieses Buchs steht: „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.“ Doch bald nachdem die erste Auflage gedruckt war, wurde klar: Es geht im Grunde gar nicht mehr darum, große Projekte agil durchzuführen. Heute machen sich ganze Organisationen auf, agiler zu werden. Krisen wie COVID-19, der immer stärker spürbare Klimawandel und geopolitische Konflikte wie der Krieg in der Ukraine haben so massiv auf die Organisationen eingewirkt, dass sie sich radikal wandeln müssen.

Die Hyperkomplexität hat mittlerweile jeden Bereich der Wirtschaft erfasst. Unternehmenslenker:innen sind dazu gezwungen, ihre Organisationen schneller als jemals zuvor mit den verfügbaren Managementmethoden an die Sachzwänge anzupassen. Längst ist erkannt, dass Organisationen, die noch immer zu viel Management mit sich herumschleppen, diese Belastung nicht mehr stemmen können. Nicht das Geld ist das Problem: Es kostet einfach zu viel Zeit. Entscheidungen dauern auf allen Ebenen zu lange, egal wo man hinschaut: Sei es die Bewilligung von Windrädern oder die Umstellung der Lieferketten, Verhandlungen mit Lieferanten oder das Ausprobieren neuer Ideen im Produktionsprozess.

Gleichzeitig gibt es jene *Enfants terribles*, wie Elon Musk, die ihre Organisationen konsequent anders aufstellen. Sie nutzen die Gesetzmäßigkeiten, die wir in diesem Buch beschreiben, statt sie zu ignorieren. Xavier Huillard teilt den Konzessions- und Baukonzern Vinci konsequent in kleine Einheiten auf: Ihm ist bewusst, dass Komplexität dann reduziert wird, wenn kleinere Einheiten selbstständig agieren können. Sogar dem Luxuslabel Hermès ist es gelungen, auf Fülle statt auf Effizienz zu setzen. In diesen Beispielen zeigt sich, dass heute das Management von Organisationen anders gedacht werden muss, wenn sie resilienter werden sollen. Um diesem Umstand Rechnung zu tragen, haben wir die zweite Auflage dieses Buchs um einige Ideen erweitert, die erfrischender sind als der ewige Ruf nach Managementmodellen, die es richten sollen.

Die Prinzipien, nach denen Sie Ihre Organisation einrichten müssen, sind sehr einfach. Überlegen Sie, ob die Organisation die geeignete Architektur hat. Ist sie in kleinen Einheiten organisiert? Nein – dann bauen Sie sie um. Erlaubt die aktuelle Informations- und Entwicklungsinfrastruktur, dass Teams Ende-zu-Ende liefern können? Nein – dann investieren Sie in den nächsten Jahren intensiv in diese Fähigkeit. Wird beständig in die Ausbildung der Mitarbeiter:innen in modernsten Arbeitsmethoden und neuesten Tools investiert? Nein – dann tun Sie es. Beginnen Sie beim Management, indem konsequent modernes Führungswissen trainiert wird. In der Produktentwicklung werden noch immer Spezifikationen geschrieben und Ihre Organisation sieht sich als Dienstleister, zum Beispiel für einen OEM? Ändern Sie Ihre Einstellung und bieten Sie Ihre Produkte mehreren Unternehmen an. Entscheiden Sie

sich für die Teilnahme an Ökosystemen, statt Lieferant zu bleiben. In Ihrer Organisation ist noch immer kein agiles Management-Framework im Einsatz? Experimentieren Sie so schnell wie möglich mit Scrum und Co.

Probieren Sie es aus. Machen Sie Ihre Erfahrungen und wenn Sie sich trauen, beginnen Sie damit in der Geschäftsleitung. Und zu guter Letzt: Wie können Sie Ihr Führungsverhalten ändern, um zuzulassen, dass Ihre Kolleg:innen ganz von selbst in die Verantwortung gehen? Möglicherweise können Ihre Kolleg:innen mehr und mehr Entscheidungen selbst treffen, während Sie sich noch stärker auf Ihre strategischen Aufgaben fokussieren.

Wir leben in spannenden Zeiten. Es ist unsere Aufgabe, Organisationen so zu bauen, dass sie den Stürmen der kommenden Jahre gewachsen sind. Dieses Buch soll Ihnen Hinweise geben, was Sie dafür tun können.

*Boris Gloger und Carsten Rasche*

Wien, Oktober 2023

# Vorwort zur 1. Auflage

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 Stegreif 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<sup>®</sup>, Nexus<sup>™</sup> 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 borisgloger 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

# Die Autoren



**Boris Gloger** ist seit 20 Jahren einer der progressivsten Denker im Bereich Management und Organisation im deutschsprachigen Raum und ein gefragter Vortragender auf Managementkonferenzen. Mit zahlreichen Publikationen, seinem Podcast und natürlich den Kolleg:innen in seinem Beratungsunternehmen bringt er das agile Denken in Organisationen verschiedenster Branchen, um deren nachhaltigen Erfolg zu sichern. Als Business Angel unterstützt Boris Gloger außerdem Start-ups dabei, ihre Ideen umzusetzen. Auf der Kundenliste von borisgloger consulting finden sich Topunternehmen aus Deutschland und Österreich.

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

- *Scrum. Produkte zuverlässig und schnell entwickeln.*  
5., überarbeitete Aufl., Carl Hanser Verlag, 2016.
- *Selbstorganisation braucht Führung. Die einfachen Geheimnisse agilen Managements.*  
3., überarbeitete Aufl., Carl Hanser Verlag, 2022.
- *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.*  
4., überarbeitete Aufl., Carl Hanser Verlag, 2023.
- *Erfolgreich mit Scrum: Einflussfaktor Personalmanagement. Finden und Binden von Mitarbeitern in agilen Unternehmen.*  
Carl Hanser Verlag, 2011.

Kontakt: [boris.gloger@borisgloger.com](mailto:boris.gloger@borisgloger.com)



**Carsten Rasche** ist seit mehr als acht Jahren Berater bei borisgloger consulting. In dieser Zeit hat er viele Business-Transformationen von Finanz- und Versicherungsunternehmen, Handelsorganisationen und Softwarekonzernen sowohl in der strategischen Ausarbeitung als auch bei der praktischen Umsetzung begleitet. Seine Expertise im skalierten agilen Arbeiten teilt er in Publikationen, auf Konferenzen und in der Trainingsreihe „myScaled Agile Practitioner“.

Folgendes Buch von Carsten Rasche ist in Zusammenarbeit mit Christoph Schmiedinger, Ellen Thonfeld und Kathrin Tuchen im Carl Hanser Verlag erschienen:

- *Agile Transformation: Der Praxisguide zum Change abseits des Happy Path.*  
2., überarbeitete Aufl., Carl Hanser Verlag, 2023

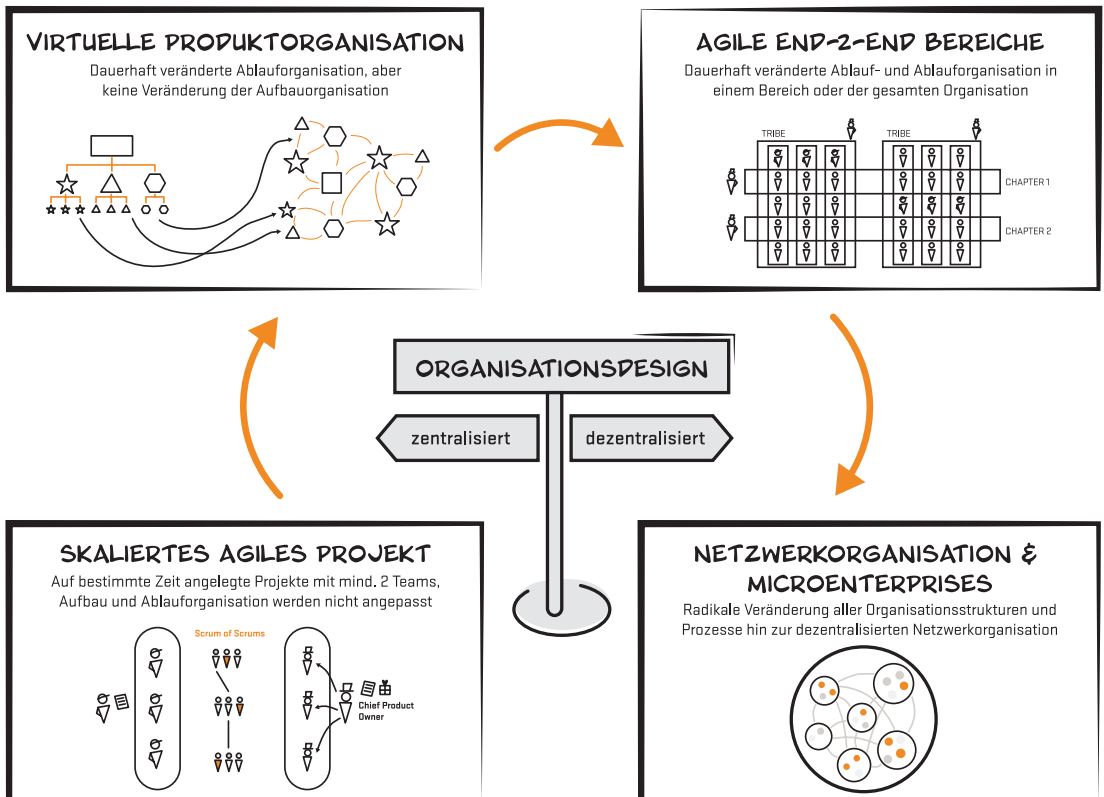
Kontakt: [carsten.rasche@borisgloger.com](mailto:carsten.rasche@borisgloger.com)



# 8

## Der Weg zur fraktal skalierten Organisation

Seit mehr als zehn Jahren begleiten wir Organisationen auf dem Veränderungsweg zu kundenorientierten und selbstorganisierten Strukturen. In den seltensten Fällen handelt es sich dabei um radikale Szenarien, in denen klassische Projekte und zentralisierte Strukturen von heute auf morgen über Bord geworfen und sofort durch dezentralisierte Netzwerke und Autonomie ersetzt werden. Aus der Beratungspraxis heraus konnten wir in den letzten Jahren im Wesentlichen vier prototypische Organisationsparadigmen ableiten, die uns im Zuge von Transformationen immer wieder begegnen (Bild 8.1).



**Bild 8.1** Vier Zielbilder, die beim Wandel zur agilen Organisation am häufigsten eingesetzt werden

Wir nennen Sie bewusst Zielbilder, da wir der festen Überzeugung sind, dass eine Organisation ein klares, von allen geteiltes Bild der angestrebten Organisationform braucht. Diese Zielbilder spiegeln auch Entwicklungsstufen wider, die Organisationen im Laufe der Jahre durchlaufen, nachdem sie klassische Aufbaustrukturen hinter sich gelassen haben.

1. Meist startet die Reise mit einzelnen Teams oder **skalierten agilen Projekten**. Für ein recht unabhängiges Vorhaben und für eine begrenzte Zeit wird agil gearbeitet und die Vorteile kleiner, autonomer Strukturen werden dabei schnell sichtbar. Dadurch entsteht der Wunsch, diese Vorteile nicht nur in abgekapselten Projekten zu nutzen, sondern sie auf größere Teile der Organisation auszuweiten.
2. Diese Ausweitung würde bedeuten, die seit vielen Jahren bestehenden Bereichs- und Abteilungsstrukturen aufzubrechen – daher wird im nächsten Schritt auf eine **virtuelle Produktorganisation** gesetzt. Teams werden multidisziplinär mit Vertreter:innen aus diversen Abteilungen zusammengesetzt, ohne die bestehende formale Organisationsstruktur und Hierarchie anpassen zu müssen. So können auch in größeren Organisationen Teams sowie Verbünde aus Teams entstehen (Team of Teams), die Produkte direkt an Kunden und Nutzer:innen ausliefern, ohne unzählige Übergaben zwischen Teams und Bereichen. Im Vergleich zu einer Projektorganisation ist hier bereits eine größere Stabilität in den Teamstrukturen und Verantwortungen erkennbar. Der dauerhafte „Verleih“ von Expert:innen in eine parallele virtuelle Produktorganisation führt jedoch zwangsweise zu Konflikten zwischen den hierarchischen Strukturen der Organisation und den Produktverantwortlichen: Welche Expert:innen werden wo zuerst eingesetzt und wer stellt das Budget zur Verfügung?
3. Der nächste konsequente Schritt ist daher die Überführung der virtuellen Strukturen in **agile End-2-End-Bereiche** (häufig Tribes oder Cluster genannt, bestehend aus maximal 125 Personen). Diese tragen die komplette – inhaltliche wie finanzielle – Verantwortung für einzelne oder mehrere Produkte und können diese autonom vom Rest der Organisation ausliefern. Gestartet wird meistens mit einzelnen Produktbereichen, langfristig wird jedoch die gesamte Organisation mit allen ihren Supportfunktionen umgebaut, um möglichst viel Autonomie in den einzelnen Bereichen herzustellen. Die übergreifende Steuerung der End-2-End-Bereiche erfolgt in der Regel noch recht klassisch über zentrale Ziele, Priorisierung und Budgetverteilung.
4. Erst der Umbau in eine **Netzwerkorganisation mit Microenterprises** führt zu einer wirklichen Verschlankung der übergreifenden Koordinationsstrukturen. Damit ist die radikale Übertragung von Verantwortung an die einzelnen Teams verbunden, die nun sowohl für Gewinn und Verlust als auch für die Entscheidungen zu den eigenen Produkten und Services zuständig sind.

In diesem Kapitel ergänzen wir die Erklärung der jeweiligen Struktur mit je einem Beispiel aus der Praxis. Aus diesen Beispielen leiten wir anschließend Designprinzipien für die einzelnen Organisationsparadigmen ab, um sie für den Umbau Ihrer eigenen Organisation nutzbar zu machen. Abschließend zeigen wir auch die Grenzen der einzelnen Organisationsparadigmen auf.

## ■ 8.1 Das skalierte agile Projekt

Richtig aufgesetzte agile Projekte können klassischen Projekten in vielen Punkten überlegen sein: Die Qualität der Ergebnisse ist meist höher, auf Veränderungen kann schnell eingegangen werden und die ausgelieferten Produktinkremente funktionieren (meistens). Daher setzen etliche Organisationen bewusst auf das agile Vorgehen und haben ihre Governance entsprechend angepasst. Uns begegnet aber auch immer wieder eine zwiespältige Herangehensweise: Die Arbeit mit agilen Rollen nach Scrum wird zwar geduldet, gleichzeitig wird aber verlangt, dass das klassische Projektcontrolling und -reporting eingehalten werden. Für die Teams bedeutet das einen erheblichen Mehraufwand, der sie von ihrer eigentlichen Aufgabe abhält: dem Liefern.

Wenn agiles Arbeiten in klassisches Projektmanagement und Vorgaben aus der Organisation gezwängt wird und auch die Zusammenarbeit mit Dienstleistern nicht adaptiert werden kann, behindert das unmittelbar den Erfolg des Projekts. Einzelne agile Praktiken, wie zum Beispiel eine gemeinsame Planung der Teams, regelmäßige Daily Standups, Review-Meetings oder Retrospektiven, machen ein Projekt nicht automatisch zu einem agilen Projekt. Ein wirklich agiles Projekt braucht die entsprechenden Voraussetzungen, damit die einzelnen agilen Teams möglichst autonom arbeiten und regelmäßig getestete Produktinkremente ausliefern können. Der Grundstein dafür wird bereits bei der Initiierung und Planung des Projekts gelegt: Schon in diesen Phasen müssen möglichst unabhängige Produktarchitekturen geschaffen, die nötige Infrastruktur – etwa für Tests und Releases – und die Skills für das Vorhaben bereitgestellt oder aufgebaut werden. Oft wird erst in mehreren Runden mit dem Management klar, dass die Teams weitreichende Entscheidungskompetenzen brauchen, damit möglichst wenige formalisierte, übergreifende Abstimmungen notwendig werden.

Sofern Steering Committees nötig sind, sollte auch dort eine Veränderung spürbar werden: Statusmeetings sollten sich zu interaktiven Arbeitssessions wandeln, in denen organisationsweite Barrieren abgebaut werden, die das schnelle Liefern behindern. Dazu gehört die gängige Praxis, Entwickler:innen gleichzeitig in mehreren Projekten und somit in mehreren Teams zu platzieren. Für das skalierte agile Projekt ist das ein No-Go, da Teams auf diese Weise die Möglichkeit genommen wird, fokussiert zu arbeiten, gemeinsam Entscheidungen zu treffen und sich völlig auf das kontinuierliche Liefern auszurichten. Allen agilen Projekten – von kleinen Vorhaben bis zu Programmen mit einer zweistelligen Anzahl von Teams – ist eines gemeinsam: Sie haben ein konkretes Projektziel und einen spezifischen Zeitrahmen, in dem etwas Bestimmtes geliefert werden soll. Projektteams werden daher für Zeiträume von einem halben Jahr bis zu zwei Jahren aufgebaut und sollten in einer konstanten Besetzung bleiben.

Ein Beispiel dafür ist die Commerzbank AG. Im Rahmen eines großen Strategieprogramms im Jahr 2016 setzte sich das Unternehmen das Ziel, viele Prozesse zu digitalisieren (vgl. Hessemüller 2019). Am sogenannten Digital Campus („Campus 1.0“) wurden für diesen Zweck die konzernweiten Digitalisierungsprojekte der Bank an einem Standort gebündelt. Dieser Campus ließ ein völlig neues, fokussiertes Umfeld entstehen. Die Teams in den diversen Projekten hatten gut ein Jahr Zeit, um ein MVP zu liefern. An den Start traten gemischte Teams aus Fach- und IT-Mitarbeiter:innen, die aus ihren jeweiligen Bereichen für die Dauer des Projekts entsendet wurden. Die Kunden wurden kontinuierlich einbezogen und durch die crossfunktionale Zusammenarbeit etablierte sich eine neue Lern- und Fehlerkultur.



Nach den zwei Jahren folgte anhand der gemachten Erfahrungen der nächste konsequente Schritt: Das Campus-Modell wurde auf große Teile der Zentrale ausgeweitet. Dadurch entstand eine Cluster-Lieferorganisation, die als feste und gleichberechtigte Struktur neben Bankfunktionen wie dem Vertrieb besteht. Das Beispiel der Commerzbank zeigt im Großen, wie die Erfahrungen aus skalierten agilen Projekten als Grundlage genutzt werden können, um eine Organisation langfristig zu verändern.

Im folgenden Abschnitt zeigen wir an einem etwas kleineren Beispiel, welche Rahmenbedingungen das skalierte agile Projekt braucht und wie es gestartet wird.

### Praxisbeispiel: Einführung eines modernen Customer Relationship Managements bei einer Versicherung

In einem weltweit tätigen Versicherungsunternehmen sollte erstmals eine Software für das Customer Relationship Management eingeführt werden. Das Ziel war, die Kunden besser zu betreuen und die Mitarbeitenden bei ihrer täglichen Arbeit zu unterstützen. Gleichzeitig sollten durch die Einführung des Softwaresystems in der Organisation die Digitalisierung und Automatisierung vorangetrieben werden. Um die Bedeutung des Vorhabens zu unterstreichen, wurde ein unternehmensweites Programm zur Steigerung der Kundenzufriedenheit ins Leben gerufen, das von zwei Bereichsleitern vorangetrieben wurde. Diese hatten auch die Hauptverantwortung für die Einführung der neuen Softwarelösung.

Als sie das Projekt aufsetzten, achteten die verantwortlichen Personen darauf, den umzusetzenden Teams möglichst viel Autonomie einzuräumen. Auf Ebene der **Organisationsstruktur und Produktarchitektur** wurde daher darauf geachtet, dass die Systemeinführung ein abgegrenztes Vorhaben blieb. Zunächst sollte das neue System im Bereich der Schaden- und Unfallversicherung eingeführt werden, bevor es auf weitere Bereiche ausgeweitet würde. Die in das Projekt involvierten Personen hatten in der Projektvorphase verschiedene Softwarelösungen begutachtet und die Lösung mit dem besten Funktionsumfang für die Bedürfnisse der Aktuarien (wissenschaftlich ausgebildete Sachverständige im Versicherungswesen, die sich mit der Bewertung und Steuerung von Risiken befassen) und die beste Integrierbarkeit in die bestehende Systemlandschaft befunden. Die Projektmitglieder konnten völlig frei entscheiden, wie die Zielarchitektur der neuen Lösung aussehen sollte und wie sie die Anbindung an die bestehenden Systeme der Organisation gestalten wollten. In den funktionalen Bereichen der Organisation wurden für das Projekt keine Änderungen vorgenommen, vielmehr wurden die benötigten Expert:innen aus den Fachbereichen und der IT für einen definierten Zeitraum zu 100 Prozent für das Projekt freigestellt. Von Beginn an wurde außerdem festgelegt, dass die Projektmitglieder in kleinen, autonomen und crossfunktionalen Teams aus drei bis neun Personen arbeiten sollten.

Auf Ebene der **Infrastruktur** war von Anfang an klar, dass die gewählte Lösung auf jeden Fall Continuous Integration, Continuous Deployment und Continuous Delivery unterstützte. Auf Seite des Dienstleisters, der die Implementierung begleitete, waren aber noch nicht alle Teammitglieder mit diesen Tools und Vorgehen vertraut. Daher wurde in den ersten Monaten ein Experte hinzugezogen, der beim Einrichten der geeigneten Toolchain half und die Entwickler:innen in den ersten Sprints dabei begleitete, diese korrekt zu nutzen.

Innerhalb der Organisation gab es zum Zeitpunkt der Einführung der neuen Lösung keine Entwickler:innen und Architekt:innen, die mit dem neuen System vertraut waren. In puncto **Skills und Professionalität** führte daher kein Weg an einem Implementierungspartner vor-

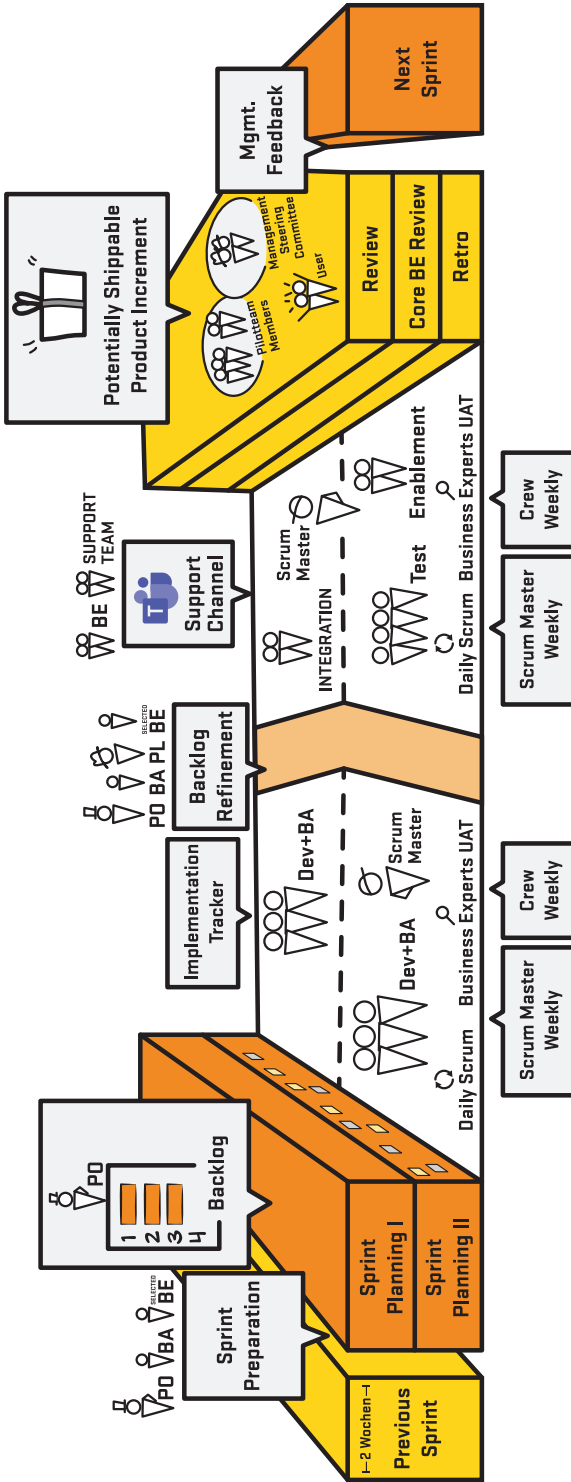
bei, der das notwendige Know-how mitbrachte. Bei der Auswahl war somit ein Kriterium, dass der Dienstleister die hauseigenen Expert:innen im neuen System schulen konnte. Dennoch wurde vor Beginn der Implementierung mit der Rekrutierung zusätzlicher Entwickler:innen begonnen, um in naher Zukunft in der Lage zu sein, den Betrieb und die Weiterentwicklung des Systems selbst zu übernehmen. Ein Top-Aktuar sollte für die Rolle des Product Owners freigespielt werden, um so das nötige Know-how in die Entwicklung zu bringen. Gleichzeitig wurden wir damit beauftragt, während der Implementierung zwei Scrum Master und einen Agile Coach zu schulen und während der ersten neun Monate im Arbeitsalltag zu begleiten.

Hauptnutzergruppe des neuen Systems waren die Aktuar. Die **Produktentwicklung** hatte sich daher das Ziel gesetzt, diese von Beginn an eng einzubinden. Zusätzlich zu dem bereits erwähnten Product Owner aus der Schaden- und Unfallversicherung wurde eine Core-Business-Expert:innengruppe aus verschiedenen Bereichen zusammengestellt, die zu etwa 20 Prozent ihrer Zeit an dem Projekt mitarbeiten sollten. Diese Gruppe unterstützte den Product Owner bei der Spezifizierung der Requirements und sie stand allen Teammitgliedern in einem Channel von Microsoft Teams für direkte Fragen zur Verfügung. Die Gruppe war außerdem fester Bestandteil der alle zwei Wochen stattfindenden Reviews, um den Teams direktes Feedback zu geben und nahe an der Entwicklung zu bleiben. Vor allem mit dem Dienstleister gab es zu Beginn intensive Diskussionen darüber, wie Kernfunktionalitäten und Module konfiguriert und angebunden werden mussten, bevor die ersten Bereiche eine Basisversion in ihrer täglichen Arbeit nutzen konnten. Man einigte sich auf das Ziel, dass nach vier Sprints à zwei Wochen eine erste nutzbare Version mit echten Kundendaten vorliegen sollte.

Was die Ebene der **Management-Frameworks** betrifft, so wollte die Organisation das Vorhaben als Leuchtturmprojekt nutzen. Es sollte deutlich werden, welche Vorteile die Arbeit in crossfunktionalen Scrum-Teams gegenüber klassischen Projektteams hat und wie die Fachbereiche durch die iterative Entwicklung besser eingebunden werden konnten. Schnell zeichnete sich ab, dass anfänglich zwei Entwicklungsteams zu je sieben bis acht Personen notwendig waren. Außerdem wurde ein Product Owner ausgewählt, der bereits in der Vorphase des Projekts eine treibende Rolle bei der Auswahl der Software und der Dienstleister gespielt hatte.

Auf Basis dieser Voraussetzungen schlugen wir vor, das Projekt sowie die Zusammenarbeit zwischen den Teams und mit den Stakeholdern anhand des LeSS-Frameworks zu strukturieren. Eine geeignete Product-Owner-Persönlichkeit stand ja bereit und die beiden Teams wurden so besetzt, dass sie als unabhängige Feature-Teams die jeweils nächsten Anforderungen aus dem Product Backlog ziehen konnten. Geholfen hatte dabei sicher, dass die Teams erst zu Projektstart gebildet worden waren und sich die einzelnen Teammitglieder daher noch nicht auf einzelne Komponenten bzw. Module der Anwendungen spezialisiert hatten.

In der Startphase und während der ersten Sprints ergänzten wir das Vorgehen nach LeSS um Elemente, die für den Kontext der Versicherung und in der Zusammenarbeit mit dem Dienstleister benötigt wurden. Das Ergebnis ist in Bild 8.2 zu sehen. Die bestehenden Strukturen wurden während der Projektlaufzeit kontinuierlich weiterentwickelt und angepasst.



**Bild 8.2** Das Management-Framework der Versicherung – LeSS, erweitert um individuelle Elemente

- In einem **Sprint-Preparation Meeting** nahm der Product Owner gemeinsam mit ausgewählten Business Experts und dem Lead Business Analyst des Dienstleisters die finale Priorisierung der User Storys für den nächsten Sprint vor und es wurden die letzten fehlenden Akzeptanzkriterien ergänzt.
- Während des Sprints konnten die Teams über einen **Microsoft Teams Support Channel** für tiefgehende fachliche Fragen in schnellen und direkten Kontakt mit den sechs Business Experts treten.
- Ein **Multi-Team Product Backlog Refinement**, in dem in der Regel alle Teammitglieder sowie die Business Experts anwesend waren, etablierte sich schnell als zentrales Element für die Ausdifferenzierung der anstehenden User Storys.
- Zusätzlich wurde ein **Scrum Master Weekly** für den Austausch zwischen den Scrum Mastern und die Bearbeitung von Impediments eingeführt. In einem weiteren Weekly ging es hauptsächlich um die Koordination mit dem Dienstleister.
- Zu dem alle zwei Wochen stattfindenden **Review** wurden von Beginn an alle Akteure der Organisation eingeladen, um sie vom Start weg auf das Projekt und die Einführung des neuen Systems aufmerksam zu machen. An dem 45-minütigen Review-Termin nahmen immer rund 200 Personen teil, die ihr Feedback geben konnten. In dieser großen Runde und in diesem Zeitrahmen war allerdings für detailliertes Feedback an die Entwickler:innen zu spezifischen Fragestellungen keine Zeit. Deshalb fand vor der großen Review-Runde jeweils ein zweistündiges Core Business Expert Review statt, in dessen Rahmen sich die sechs Business Experts in Ruhe durch die neuen Funktionalitäten klicken konnten. Im Sinne des Erwartungsmanagements für die Organisation wurde zusätzlich ein Implementation Tracker eingeführt, über den sich die Akteure darüber informieren konnten, wann die Implementierung weiterer neuer Funktionalitäten geplant war.
- Nach den beiden Review-Terminen fand zusätzlich eine **Management-Feedbackrunde** statt, in der die beiden projektverantwortlichen Bereichsleiter mit dem Product Owner und dem zentralen Ansprechpartner des Dienstleisters zusammenkamen. Im Rahmen dieses Meetings wurden etwaige spezifische Rückfragen aus dem Review-Termin beantwortet, die Priorisierung für die kommenden Sprints wurde abgestimmt und es wurden Lösungen für Impediments gesucht, die nicht innerhalb des Projekts gelöst werden konnten. Alle zwei Monate wurde der Projektlenkungsausschuss mit den verantwortlichen Top-Führungskräften in dieses Meeting integriert.
- Für beide Teams fand eine gemeinsame **Retrospektive** statt, da die meisten Impediments teamübergreifender Natur waren.

Auf Ebene der **Führung und Werte** wurde darauf geachtet, dass keine direkte disziplinarische Führungskraft einzelner Personen auch Mitglied desselben Teams war. Der Dienstleister setzte sich dafür ein, dass weisungsbefugte Rollen wie die Projektleitung ebenfalls nicht direkte Mitglieder der Teams waren, sondern eine koordinierende Rolle einnehmen konnten. Um die Selbstorganisation und das Übernehmen von Verantwortung seitens der Teammitglieder zu stärken, gaben die Scrum Master in den ersten Sprints kurze Inputs zu diesen Themen und es wurden kleine Experimente durchgeführt.

Es dauerte ein gutes halbes Jahr, bis die meisten Grundfunktionalität des Systems implementiert waren. Die beiden Teams hatten danach noch ein gut gefülltes Backlog, um weitere Zusatzfunktionen einzubauen. Zusätzlich wurde ein weiteres Team mit einem weiteren Product Owner gegründet, das einen spezifischen Angebotsprozess für Lebensversicherun-

gen abdeckte. Da von Anfang an ein entkoppelter Ansatz verfolgt worden war, gelang diese Erweiterung problemlos. Ohne langes Onboarding und komplizierte Abstimmungen konnte dieses Team sofort seine Arbeit aufnehmen.

### 8.1.1 Designprinzipien für das skalierte agile Projekt

Das Beispiel zeigt auf, auf welche Elemente bei einem agilen Projekt geachtet werden sollte, um es zum Erfolg zu führen. Die zentralen Designprinzipien haben wir in Tabelle 8.1 für jede Ebene der Skalierungspyramide zusammengefasst, damit Sie diese als Grundlage für ihr eigenes Projektsetup nutzen können. Diskutieren Sie in Ihren Projektteams, wie Sie diese Prinzipien in Ihrem Kontext berücksichtigen können und welche Veränderungen dazu in der Organisation angestoßen werden müssen. Diese Prinzipien eignen sich auch dafür, bestehende Projektsetups zu untersuchen und weiterzuentwickeln.

**Tabelle 8.1** Designprinzipien für ein skaliertes agiles Projekt. Die kursiv hervorgehobenen Prinzipien sind spezifisch für ein skaliertes agiles Projekt.

Ebene	Ausprägung
<b>Führung &amp; Werte</b>	<ul style="list-style-type: none"> <li>▪ <i>Disziplinarische Führungsrollen verbleiben in der bestehenden Organisationsstruktur.</i></li> <li>▪ Projektleitung und Management führen lateral und schaffen die Rahmenbedingungen, damit Teams optimal liefern können.</li> <li>▪ Jede:r Einzelne übernimmt Verantwortung, sowohl für fachliche Inhalte als auch die organisatorische Gestaltung des Projekts.</li> <li>▪ Um den fachlichen Diskurs und die Entscheidungsfindung zu fördern, gibt es keine disziplinarische Führung und Weisungsbefugnis innerhalb eines Teams.</li> </ul>
<b>Management-Frameworks</b>	<p><b>Steuerung</b></p> <ul style="list-style-type: none"> <li>▪ <i>Die Projektsteuerung findet in dafür vorgesehenen Reviews, Plannings und Refinements statt und nicht in separaten Steering-Committee-Meetings.</i></li> <li>▪ <i>Agile Artefakte (priorisiertes Backlog, aktuelle Roadmap und Velocity-Metriken) ersetzen das klassische Projektreporting.</i></li> </ul> <p><b>Rollen</b></p> <ul style="list-style-type: none"> <li>▪ Eine Person ist verantwortlich für die Priorisierung der Geschäfts- und Kundenanforderungen sowie den Return on Investment von einem oder mehreren Teams (maximal vier).</li> <li>▪ Eine Person pro Team ist verantwortlich für die produktive Zusammenarbeit im Team und zwischen den Teams.</li> </ul>
<b>Produktentwicklung</b>	<ul style="list-style-type: none"> <li>▪ Das Anforderungsmanagement findet innerhalb des jeweiligen Teams statt.</li> <li>▪ Regelmäßige Zusammenarbeit mit Kunden und Endnutzer:innen</li> <li>▪ Produktinkremente werden hypothesengetrieben entwickelt (Minimum Viable Product) und direkt durch die Nutzer:innen validiert.</li> </ul>

**Tabelle 8.1** (Fortsetzung) Designprinzipien für ein skaliertes agiles Projekt. Die kursiv hervorgehobenen Prinzipien sind spezifisch für ein skaliertes agiles Projekt.

Ebene	Ausprägung
<b>Skills &amp; Professionalität</b>	<ul style="list-style-type: none"> <li>▪ Im Team sind das fachliche Wissen und das Domänen-Know-how vorhanden.</li> <li>▪ Im Team ist das methodische Know-how (bspw. Entwicklungsmethoden) vorhanden.</li> <li>▪ Die Teammitglieder besitzen die Flexibilität, kreative Lösungen für technische und organisatorische Probleme zu entwickeln.</li> </ul>
<b>Infrastruktur</b>	<ul style="list-style-type: none"> <li>▪ <i>Teams können den Projektscope bzw. einen Teil davon ohne den Beitrag anderer Teams eigenständig entwickeln und liefern.</i></li> <li>▪ Kontinuierliches, automatisiertes und teamunabhängiges Testen einzelner Produktinkremente ist möglich.</li> </ul>
<b>Struktur &amp; Architektur</b>	<p><b>Technische Architektur</b></p> <ul style="list-style-type: none"> <li>▪ <i>Das Vorhaben für das Projekt ist abgegrenzt und die Architektur ist entkoppelt.</i></li> <li>▪ <i>Die Entwicklungsteams können die Architektur selbst bestimmen.</i></li> </ul> <p><b>Organisationsarchitektur</b></p> <ul style="list-style-type: none"> <li>▪ <i>Bestehende funktionale Bereiche der Organisation bleiben bestehen.</i></li> <li>▪ <i>Die Projektmitglieder werden für einen bestimmten Zeitraum entsendet.</i></li> </ul> <p><b>Teamarchitektur</b></p> <ul style="list-style-type: none"> <li>▪ Kleine, autonome und crossfunktionale Teams bestehend aus 3–9 Personen, zusammengesetzt aus Expert:innen unterschiedlicher Bereiche</li> <li>▪ Teammitglieder gehören zu genau einem Team und arbeiten mit 100 % ihrer Kapazität in diesem Team.</li> <li>▪ Bei geändertem Projektbedarf werden sowohl die Zusammensetzung der Teams als auch deren Verantwortung angepasst.</li> </ul>



### Designprinzipien als Leitplanken für die Organisationsentwicklung nutzen

Kein agiles Arbeiten ohne Prinzipien! Generell beschreiben Prinzipien die Grundsätze und Gesetzmäßigkeiten, nach denen wir unser Handeln ausrichten. Wenn es um die Veränderung der Organisation geht, helfen Designprinzipien dabei, die grundlegenden Gestaltungsmerkmale für agile Organisationen zu beschreiben.

Wir nutzen die Prinzipien, um die diversen Impulse der vorangegangenen Kapitel in kurzen und prägnanten Sätzen für jede Ebene der Skalierungspyramide zusammenzufassen. Dadurch werden die wichtigsten Gestaltungselemente mit denjenigen Personen diskutierbar, die den Organisationsumbau vorantreiben.

### 8.1.2 Grenzen des skalierten agilen Projekts

IT-Projekte sind für viele Organisationen ein gutes Einfallstor, um das agile Arbeiten in der Organisation erstmalig auszuprobieren. Sofern das Ziel aber die fraktal skalierte agile Organisation ist, stößt dieser Ansatz schnell an seine Grenzen. Die größten Herausforderungen, die wir in der Praxis beobachten, sind folgende:

- Projekte werden immer auf eine **begrenzte Zeit** ausgelegt. Irgendwann werden die Arbeitsergebnisse in die regulären, klassisch arbeitenden Organisationseinheiten und Teams übergeben, damit die Anwendung betrieben und weiterentwickelt werden kann. Die inhaltliche Verantwortung wandert dabei meistens in den Fachbereich zu spezifischen Ansprechpartner:innen, die technische Verantwortung übernimmt die IT. Vor allem während des Übergabeprozesses geht viel Wissen aus der ursprünglichen Entwicklung verloren, wenn die Projektmitglieder danach das nächste Projekt starten. Neue Anforderungen können nicht mehr so schnell umgesetzt und Fehler nicht so schnell behoben werden, weil nicht mehr das bereits eingespielte Projektteam daran arbeitet. Änderungen und das Beheben von Fehlern am Produkt müssen von nun an in einem offiziellen Demandprozess bei der IT angemeldet werden.
- Neue Projekte starten in der Regel immer **innerhalb der bestehenden Architektur und Infrastruktur** der Organisation. Sofern nicht bereits Microservice-Architekturen sowie die geeigneten Release- und Test-Infrastrukturen vorhanden sind, müssen diese zu Projektbeginn erst einmal aufgebaut werden. Da viele Projekte von Beginn an unter einem enormen Liefer- und Zeitdruck stehen, wird häufig zu wenig darin investiert und die Teams müssen sich mit Workarounds und manuellen Tests zufriedengeben, um schnelle Fortschritte erzielen zu können. Hinzu kommt, dass das Projektteam nach Projektabschluss nicht für das Betreiben und die Weiterentwicklung des Produkts verantwortlich ist, was die langfristige Betreiberperspektive einschränkt.
- Zu Beginn eines neuen Projekts wird in der Regel ausgehandelt, welchen **Umfang** es hat und auf welche Personen das Projekt in welchem Ausmaß zugreifen darf. Ändert sich der Bedarf, sind häufig viele Gespräche und aufwendige Vereinbarungen notwendig, um daran etwas zu verändern. Parallele Projektstrukturen verhindern daher häufig fluide Strukturen, die schnelle Teamwechsel möglich machen.
- **Die bestehenden Governance- und Controlling-Strukturen** in einer Organisation sehen selten selbstorganisierte und agil arbeitende Teams vor, die schnell weitreichende Entscheidungen treffen dürfen. In der Regel müssen die bestehenden, aufwendigen Priorisierungs- und Entscheidungsrounds, Steering Committees und Reporting-Prozesse eingehalten werden. Das passt nicht mit dem Treffen schneller Entscheidungen und dem Finden von kreativen Lösungen in der Produktentwicklung zusammen.
- Die Mitarbeiter:innen einer Organisation arbeiten häufig **nur für einen bestimmte Zeitraum** in den selbstorganisierten Strukturen, wo sie ihre Potenziale voll einbringen können. Danach kehren sie wieder in die klassisch agierende Linienorganisation zurück. Das Potenzial der Mitarbeiter:innen bleibt ungenutzt und oft wollen die Kolleg:innen danach nicht wieder in den klassischen Strukturen arbeiten.

# Stichwortverzeichnis

## A

Abhängigkeiten 68, 89, 173  
Achtsamkeit 116, 126  
Agile End-2-End-Bereiche 258  
Agile Manifesto 20, 215  
Agile Release Train 247, 251  
Aktionsforschung 22  
Änderungsrate 20  
Anforderungsmanagement 109, 132, 133  
– agiles 135  
Anwender 155  
Arbeitsrecht 95  
Arbeitsschutzgesetz 14  
Architektur 39  
– emergente 43  
Artefakte 186  
Auslastung 109, 199

## B

Backlog 46, 203, 207  
Batchsize 193  
Beschleunigung 44  
Betrieb 64  
BizDevOps 90  
Burn-out 16  
Bürokratie 27  
Business-Analyse 132  
Business Model Canvas 195

## C

Change-Team 201  
Chaträume 77  
Complicated-Subsystem Team 248  
Cone of Uncertainty 25  
Constraints 46, 108, 155, 157, 192, 198  
Conway's Law 39, 54  
Cost of Delay 164, 167, 197  
Crossfunktionalität 56, 182

## D

Daily Scrum 111, 170  
Defect Rate 204  
Designprinzipien 255, 266  
– skaliertes agiles Projekt 244  
Design Thinking 5, 23, 133, 139  
Deskalisierung 144, 153  
DevOps 89, 90, 149  
Dialog 62  
Digitalisierung 10  
Discovery 140  
Disziplin 102, 169  
Durchfluss 53, 136, 194, 199  
Durchlaufzeit 189, 199  
Durchsatz 109, 167

## E

Einkauf 89  
Einladung 182, 228  
Empowerment 222  
Enabling Teams 249  
End-2-End-Bereich  
– agiler 238  
– Designprinzipien 266  
Engpass 108, 165  
Entkopplung 47  
Entwicklungspraktiken, agile 111  
Entwicklungsteam 136, 158, 160  
– Skills 119  
Entwicklungstools 85  
Estimation Meeting 170  
– skaliertes 179  
Experiment 224  
Exploration 143  
Extreme Manufacturing 13  
Extreme Programming 70, 110



**F**

Facilitation 169  
 Fähigkeiten 225  
 – soziale 102  
 Features 165  
 Feature-Team 184  
 Feedback 170, 223  
 FIFO 197  
 Flexibilität 15  
 Flipcharts 76  
 Fokus 108  
 Fokusgruppen 202  
 Freiwilligkeit 170  
 – Prinzip der 182  
 Fremdarbeitskräfteregelung 95  
 Führung 182, 210, 213  
 – disziplinarische 157  
 – laterale 158  
 Funktionalität 51, 181

**G**

Generalistentum 7  
 Gesamtprofitabilität 189  
 Gesetz der zwei Füße 183, 230  
 Gesundheit 105  
 Gilden 62, 77, 180, 181  
 Gruppendruck 16

**H**

Handover 7, 88, 92, 187  
 Holacracy 12  
 Host einer Gilde 182  
 Hyperspezialisierung 3, 6  
 – interorganisationale 7  
 – intraorganisationale 7

**I**

Ideation 141  
 Impediment 109, 174  
 Informationsstruktur 41  
 Infrastruktur 67  
 – bauliche 72  
 Integration 67  
 Interaktion 81  
 Interesse, intrinsisches 212

**J**

Journal 116

**K**

Kadenz 169  
 Kollaborationstools 78  
 Kommunikationsfähigkeit 102  
 Kommunikationsstruktur 40  
 Kommunikationstools 77  
 Komplexität 62  
 Kontrolle 27, 29, 191, 192, 200, 222  
 KPI 203  
 kritischer Pfad 165  
 Kultur des Gelingens 125, 225  
 Kundenzufriedenheit 204

**L**

Landeskulturen 126  
 Leadership 113, 211  
 Leading Target 129  
 Lean Management 5, 9, 108, 193  
 Lean Manufacturing 193  
 Lean Product Development 194  
 Lean Production 5  
 Lean Startup 145  
 Leistungsfähigkeit eines Systems 108, 114  
 Leitplanken 155  
 Lerntheorie 99  
 LeSS 153  
 Licht 74  
 Lieferanten 41, 63  
 – Zusammenarbeit mit 88  
 Liefergeschwindigkeit 189

**M**

Maker-Szene 86  
 Management 17  
 – agiles 9, 124  
 – Skills 121  
 Marktchancen 26  
 Material 74  
 Medizintechnik 94  
 Meetinginflation 8  
 Meetingräume 75  
 Meetings 169  
 Messbarkeit 30  
 Microenterprise 238, 270  
 – Designprinzipien 279  
 Microservices 51, 224  
 Mikromanagement 191  
 Minimum Viable Product 145, 244, 256, 267  
 Misstrauen 191  
 Mitarbeiterzufriedenheit 204

Mob Programming 32, 59, 102  
 Moderation 173  
 Mushroom-Architektur 40

## N

Neinsagen 181  
 Net-Promoter-Score 204  
 Netzwerke 217  
 Netzwerkorganisation 33, 238, 270  
 – Designprinzipien 279  
 Neurolinguistische Programmierung 214  
 NexusTM 153  
 Nichtwissen 103  
 Normen 52

## O

Obeya-Raum 157, 178  
 Offenheit 224  
 One Piece Flow 108, 172, 193  
 Open Space Technology 226  
 Organisation  
 – agile 201  
 – fraktal skalierte 123, 237  
 – gelingende 209  
 – teambasierte 2  
 Output 189

## P

Pairing 59  
 Parallelstrukturen 21  
 Performance-Indikatoren 203  
 Pinnwände 74  
 Planung 224  
 Platform Team 248  
 Plattform 52  
 Policies 92  
 Portfoliomanagement 188  
 – agiles 153, 188  
 – klassisches 190  
 Practice Groups 181  
 Priorisierung 194  
 Product Backlog 130  
 Product Owner 110, 136, 164, 168  
 – Skills 116  
 Product Owner Daily 175  
 Product-Owner-Team 181  
 Product Owner Weekly 177  
 Produktarchitektur 181  
 – agile 50  
 Produktentwicklung 22, 129

– neue Formen der 109  
 Produktentwicklungsprozess, agiler 149  
 Produktentwicklungszyklus 19  
 Produktidee 133  
 Produktindikatoren 204  
 Produktinkrement 130  
 Produktivität 30, 103, 104, 109, 157, 189  
 Produktivitätssteigerung 42  
 Produktorganisation, virtuelle 238, 247  
 Professionalität 18, 97  
 Programmvision 157  
 Projekt  
 – agil skaliert 239  
 – skaliertes agiles 238  
 – verteiltes 85  
 Projektmanagement 3, 19  
 Projektmanagement-Office 156, 184, 194  
 – agiles 200  
 Projektmanager 168  
 Projektplanung 25  
 Projektteam 158  
 Prototyp 143  
 Puffer 198  
 Pull-Prinzip 80  
 Purpose 216

## Q

Qualität 112  
 Qualitätssicherung 44

## R

Rahmenbedingungen 155, 181  
 Raumsituation 72  
 Redundanz 35, 48, 53  
 Refactoring 52, 192  
 Regenschirmeffekt 124  
 Regularien 94, 112, 155  
 Reporting 79, 157, 203  
 Respekt 120  
 Responsive Manifesto 216  
 Retrospektive 84, 115, 170  
 Return on Investment 8, 109, 163, 168, 194  
 Review 170, 204  
 Richtlinien 92  
 Rollen 122, 180  
 Rollenverständnis 110

## S

SAFe® 153  
 Saxonia Systems AG 80

- Schnittstelle 40, 52
  - Scrum 2, 110, 201
  - Scrum-Board 173
  - Scrum Master 104, 157, 201
    - Skills 112
  - Scrum Master Daily 175
  - Scrum-Master-Team 180
  - Scrum Master Weekly 176
  - Scrum of Scrums 171, 173
  - Selbsteinschätzung 103
  - Selbstführung 116
  - Selbstmanagement 102
  - Selbstreferenz 15
  - Selbstreflexion 115
  - Shu Ha Ri 226
  - Skalierung, fraktale 207
  - Skalierungstoolbox 168
  - Skills 16, 23, 97
    - der Organisation 17
    - des Einzelnen 101
    - des Individuums 17
  - Slack 166, 192, 199
  - Small World Network 34, 48
  - Spezialisten 7, 23
  - Spotify-Modell 259
  - Sprint Planning 1 170
  - Sprint Planning 2 170
  - Stage-Gate-Prozess 24
  - Status 100
  - Steering Committee 156, 239
  - Steuerung, zentrale 154
  - Story Map 177
  - Strategieentwicklung, agile 81
  - Stream-Aligned Team 248
  - Struktur, formale 231
  - Support-Teams 183
  - Swarming 32
  - Synchronisation 80
  - System, fraktales 123, 208
- T**
- Taskboard 15
  - Teamfähigkeit 102
  - Team of Teams 163, 238, 248, 249
  - Teams 180
  - Teampologien 248
  - Teamzusammensetzung 62
  - Technik 74
  - Technologie 62
  - Testautomatisierung 71
  - Test Driven Development 70, 110
    - in der Hardwareentwicklung 72
  - Theory of Constraints 4, 47, 108, 165, 219
  - Timebox 169
  - Toyota Production System 108
  - Transaktionskosten 209
  - Transformation Team 177, 180, 201
  - Transparenz 80, 82, 224
  - Tribe 238, 258
  - T-Shaped People 7
- U**
- Überkapazitäten 54
  - Unterstützung 103
  - User Story 130, 177
- V**
- Variabilität 194
  - Varianz 194
  - Veränderung 44, 201
  - Verantwortung 229, 230
    - der Teams 163
    - Übernehmen von 105
  - Verhalten 233
  - Vertrauen 160, 209, 222
  - Videokonferenz 77, 81
  - Vision 113, 116, 162
    - persönliche 105
  - Visualisierung 82
- W**
- Wasserfall 26
  - Wasserfallmodell 69, 109
  - Waste 193
  - Weighted Shortest Job First 198
  - Werte 192, 194
    - von Scrum 84
  - Wertesystem 211
  - Wertschöpfungskette 10
  - Whiteboard 76
  - Widerstand 100, 217, 219
  - WIP-Limit 204
  - Wissensmanagement 5
  - Wissensträger 4
  - Work in Progress 193
    - Limit 111
- Z**
- Zusammenarbeit 78
    - verteilte 81