

## Scrum Think big

Scrum für wirklich große Projekte,  
viele Teams und viele Kulturen

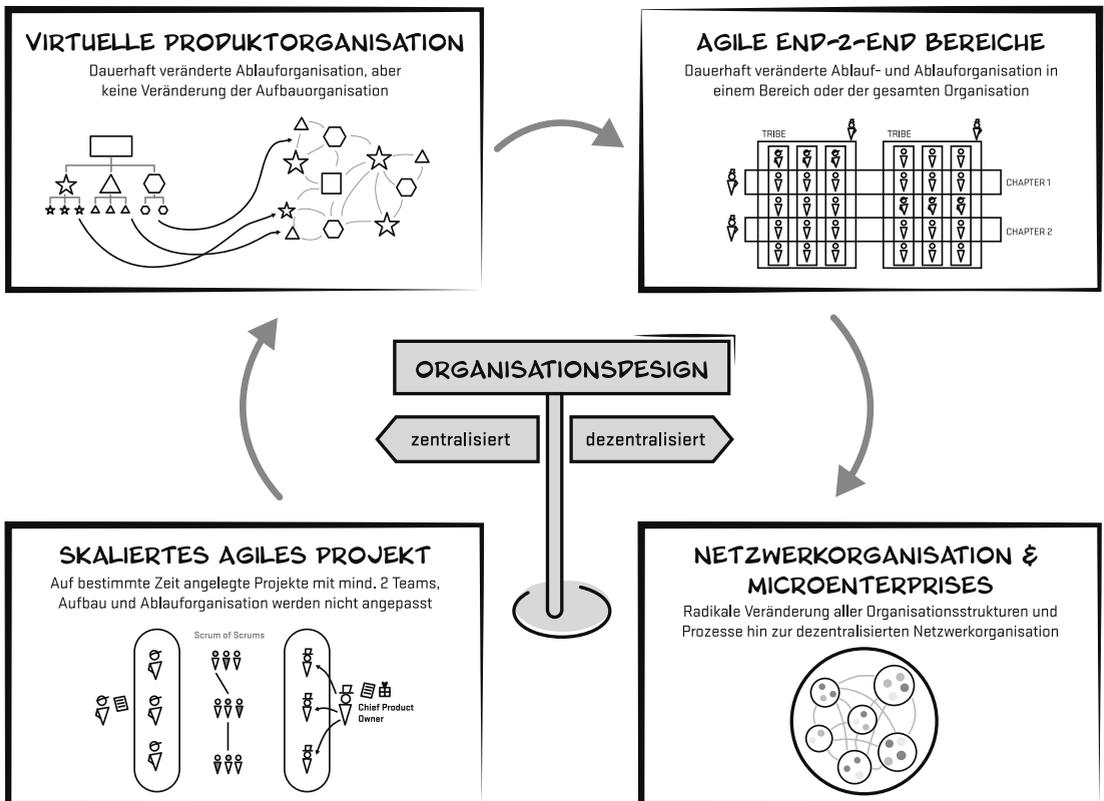
» Hier geht's  
direkt  
zum Buch

# DIE LESEPROBE

# 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-

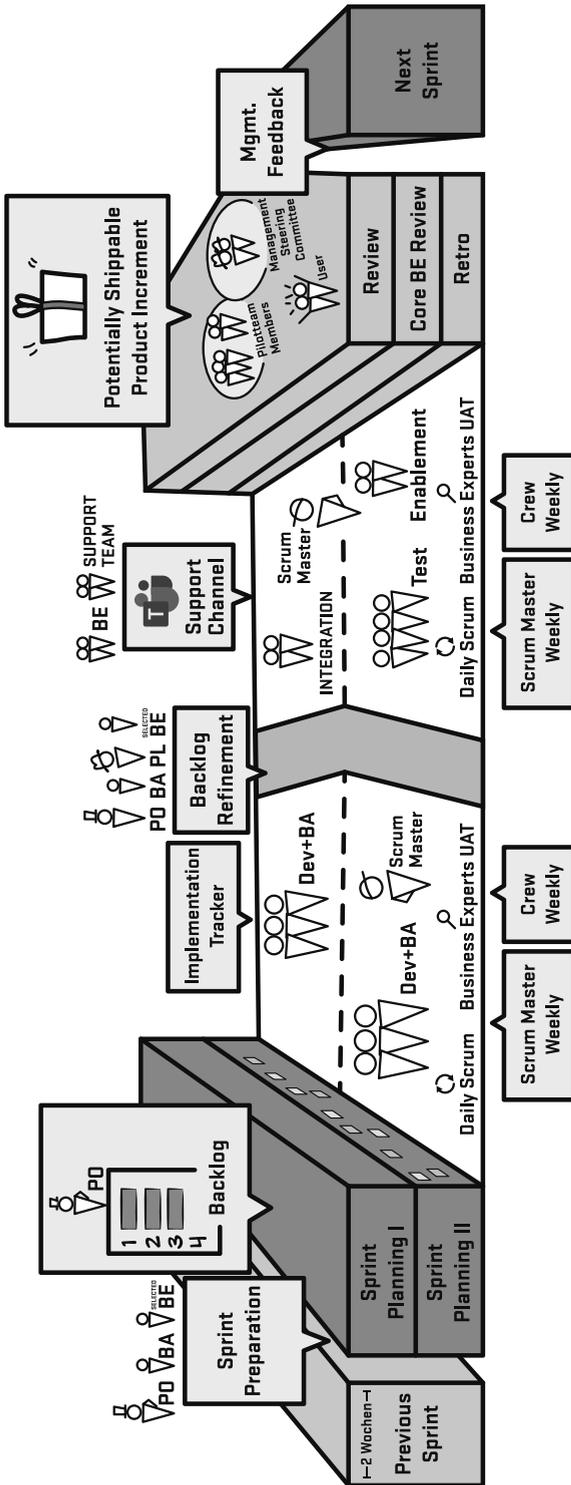
bei, der das notwendige Know-how mitbrachte. Bei der Auswahl war somit ein Kriterium, dass der Dienstleister die hauseigenen Expert:innen im neuen System schulen konnte. Dennoch wurde vor Beginn der Implementierung mit der Rekrutierung zusätzlicher Entwickler:innen begonnen, um in naher Zukunft in der 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 Aktuare. 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.