

O'REILLY®

Inklusive
Workbook zu
Interaktionen
in verteilten Teams

TEAM

TOPOLOGIES

Organisation
von Business- und
IT-Teams für einen
schnellen
Arbeitsfluss

Matthew Skelton, Manuel Pais
Übersetzung von Michael Plöd

Team Topologies

*Organisation von Business- und IT-Teams für
einen schnellen Arbeitsfluss*

Inklusive

Interaktionen in verteilten Teams – Workbook

*Team-Topologies-Patterns für eine
produktivere Zusammenarbeit*

Matthew Skelton, Manuel Pais

*Deutsche Übersetzung von
Michael Plöd*

O'REILLY®

Matthew Skelton, Manuel Pais

Lektorat: Ariane Hesse

Übersetzung: Michael Plöd

Copy-Editing: Claudia Lötschert, www.richtiger-text.de

Satz: III-satz, www.drei-satz.de

Herstellung: Stefanie Weidner

Umschlaggestaltung: Michael Oréal, www.oreal.de

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-231-5

PDF 978-3-96010-836-8

ePub 978-3-96010-837-5

mobi 978-3-96010-838-2

1. Auflage 2024

Translation Copyright © 2024 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Authorized German translation of the English original »Team Topologies: Organizing Business and Technology Teams for Fast Flow«, Copyright © 2019 by Matthew Skelton and Manuel Pais, ISBN 9781942788812; »Remote Team Interactions Workbook: Using Team Topologies Patterns for Remote Working« Copyright © 2022 by Matthew Skelton and Manuel Pais, ISBN 9781950366347. This translation is published and sold by permission of IT Revolution Press LLC, which owns or controls all rights to publish and sell the same.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:

Dieses Buch wurde mit mineralölfreien Farben auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie. Hergestellt in Deutschland.



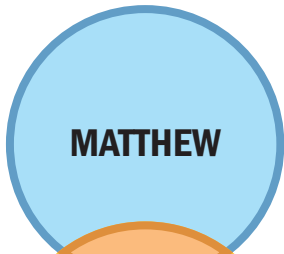
Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: komentar@oreilly.de.

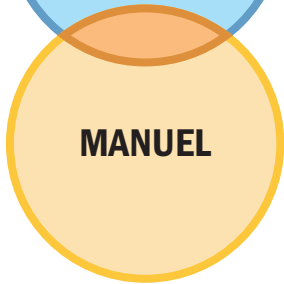
Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autoren noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.



Meiner Frau Suzy Beck – für all ihre Unterstützung und Inspiration.



Für Katie, meine Lebensgefährtin und Hochburg unserer Familie – danke für deine unermüdliche Liebe und Unterstützung.

Für Dan und Ben, die täglichen Quellen der Wärme – hoffentlich kann dieses Buch dazu beitragen, dass ihr versteht, was Daddy beruflich macht.

Team Topologies

| | |
|---|-----------|
| Vorwort | 15 |
| Vorwort des Übersetzers | 17 |
| Einleitung | 19 |
| <hr/> | |
| Teil I Teams als entscheidendes Mittel bei der Lieferung | 26 |
| Die wichtigsten Erkenntnisse | 27 |
| 1 Das Problem mit Organisationsdiagrammen | 29 |
| Kommunikationsstrukturen einer Organisation | 30 |
| Team Topologies: Eine neue Art, über Teams zu denken | 35 |
| Das Revival von Conway's Law | 36 |
| Kognitive Belastung und Engpässe | 37 |
| Zusammenfassung: Überdenken Sie Teamstrukturen, Zweck und Interaktionen | 39 |
| 2 Das Conway'sche Gesetz und seine Bedeutung | 41 |
| Das Conway'sche Gesetz verstehen und anwenden | 41 |
| Das Reverse Conway Maneuver | 44 |
| Softwarearchitekturen, die einen teamorientierten Arbeitsfluss fördern | 48 |
| Organisationsgestaltung erfordert technische Expertise | 49 |
| Beschränken Sie unnötige Kommunikation | 50 |
| Achtung! Naive Anwendungen des Conway'schen Gesetzes | 52 |
| Zusammenfassung: Conway's Law ist entscheidend für effizientes Teamdesign in der Tech-Branche | 55 |

| | |
|--|-----------|
| 3 Teamorientiertes Denken | 57 |
| Setzen Sie auf kleine, langlebige Teams als Standard | 58 |
| Gute Grenzen verringern die kognitive Belastung | 65 |
| Entwerfen Sie »Team-APIs« und gestalten Sie Teaminteraktionen | 74 |
| Warnung: Engineering-Praktiken sind unerlässlich | 83 |
| Zusammenfassung: Begrenzen Sie die kognitive Belastung von Teams und gestalten Sie Teaminteraktionen, um einfacher und schneller zu arbeiten | 83 |

Teil II Team Topologies, die für einen reibungslosen Arbeitsfluss sorgen

| | |
|--|------------|
| Die wichtigsten Erkenntnisse | 87 |
| 4 Statische Team Topologies | 89 |
| Team-Anti-Patterns | 90 |
| Design für den Fluss des Wandels | 91 |
| DevOps und die DevOps-Topologien | 93 |
| Erfolgreiche Team-Patterns | 95 |
| Überlegungen bei der Auswahl einer Topologie | 100 |
| Verwendung von DevOps-Topologien zur Weiterentwicklung der Organisation | 104 |
| Zusammenfassung: Adaptieren und entwickeln Sie Team-Topologies, die zu Ihrem aktuellen Kontext passen | 106 |
| 5 Die vier grundlegenden Team Topologies | 107 |
| Stream-aligned Teams | 109 |
| Enabling Teams | 114 |
| Complicated-subsystem Teams | 118 |
| Platform Teams | 120 |
| Vermeiden Sie Team-Silos im Zuge des Wandels | 127 |
| Eine gute Plattform ist »gerade groß genug« | 128 |
| Ordnen Sie die bekannten Teamtypen den grundlegenden Team Topologies zu | 132 |
| Zusammenfassung: Verwenden Sie lose gekoppelte, modulare Gruppen von vier spezifischen Teamtypen | 137 |
| 6 Entscheiden Sie sich für teamorientierte Grenzen | 139 |
| Ein teamorientierter Ansatz für Zuständigkeiten und Grenzen von Software | 140 |
| Versteckte Monolithen und Kopplung | 140 |
| Softwaregrenzen oder »Bruchflächen« | 143 |

| | |
|---|-----|
| Beispiel aus der realen Welt: Fertigungsindustrie | 152 |
| Zusammenfassung: Wählen Sie Softwaregrenzen, die der kognitiven Belastung des jeweiligen Teams entsprechen | 154 |

Teil III Evolution von Teaminteraktionen für Innovation und schnelle Lieferfähigkeit

| | |
|--|------------|
| Die wichtigsten Erkenntnisse | 157 |
| 7 Die Modi der Teaminteraktion | 159 |
| Gut definierte Interaktionen sind der Schlüssel zu effektiven Teams . . . | 160 |
| Die drei wesentlichen Modi der Teaminteraktion. | 161 |
| Teamverhaltensweisen für jeden Interaktionsmodus | 169 |
| Auswahl geeigneter Modi für die Teaminteraktion. | 171 |
| Auswahl der grundlegenden Teamorganisation | 172 |
| Wählen Sie Teaminteraktionsmodi, um Unsicherheiten zu verringern und den Arbeitsfluss zu verbessern | 175 |
| Zusammenfassung: Drei gut abgegrenzte Modi der Teaminteraktion . . | 178 |
| 8 Entwickeln Sie Teamstrukturen mit einem Gespür für organisatorische Belange. | 179 |
| Wie viel Collaboration ist für jede Teaminteraktion angemessen? | 179 |
| Beschleunigung des Lernens und der Übernahme neuer Praktiken | 181 |
| Konstante Evolution der Team Topologies. | 185 |
| Die Kombination von Team Topologies für mehr Effektivität | 190 |
| Auslöser für die Evolution von Team Topologies. | 191 |
| Selbststeuerung von Design und Entwicklung | 195 |
| Zusammenfassung: Evolutionäre Team Topologies | 201 |
| Schlussfolgerung: Das digitale Betriebsmodell der nächsten Generation | 203 |
| Anhang A Glossar | 213 |
| Anhang B Literaturempfehlungen. | 217 |
| Anhang C Literaturverzeichnis | 219 |
| Anhang D Danksagungen | 235 |

Interaktionen in verteilten Teams – Workbook

| | |
|--|------------|
| Vorwort | 241 |
| Einleitung | 243 |
| 1 Überblick – Fokus auf remote Teaminteraktionen | 247 |
| Was braucht ein Unternehmen, um in einer remote-orientierten Welt erfolgreich zu sein? | 247 |
| Verwenden Sie den Team-API-Ansatz zum Definieren und Kommunizieren von Verantwortlichkeiten und Team-Fokus | 250 |
| Abhängigkeiten mit einfachen Tools verfolgen und blockierende Abhängigkeiten entfernen | 250 |
| Kommunizieren Sie viel, dokumentieren Sie schriftlich aber nur das Nötigste | 251 |
| Zusammenfassung: Gestalten und definieren Sie die Art und Weise, wie Teams interagieren | 252 |
| 2 Teamabhängigkeiten | 253 |
| Team-API | 253 |
| Tracking von Abhängigkeiten | 259 |
| Netzwerke aufbauen: Kaffee, Gespräche, interne Konferenzen | 263 |
| 3 Festlegung von Teamgrenzen | 269 |
| Vertrauensgrenzen für Gruppen | 269 |
| Einrichtung des Online-Raums | 273 |
| Teamorientierte Konventionen für Chat-Tools | 279 |
| 4 Zweckmäßige Interaktionen | 283 |
| Modi der Teaminteraktion: Ein Rückblick | 283 |
| Beobachten von Teaminteraktionen | 288 |
| Klärung von Kommunikationszweck und -kanälen | 292 |
| Für die Klarheit des Zwecks von Plattformen und Services sorgen | 294 |
| 5 Nächste Schritte | 301 |
| Entwerfen und Durchführen einer Plattform-Umfrage zur Developer Experience | 301 |
| Definieren Sie Namens- und Nutzungskonventionen für Chat-Tools | 303 |
| Verwenden Sie die Team-API mit mehreren Teams, um Teamgrenzen zu definieren und zu präzisieren | 304 |
| Erstellen und teilen Sie einen Durchführungsplan | 304 |

| | |
|----------------------------------|-----|
| Liste der Ressourcen | 307 |
| Über die Autoren | 309 |
| Stimmen zu Team Topologies | 311 |
| Gesamtindex | 315 |

Team Topologies

Unsere Systeme klein und einfach zu halten, ist ein erstrebenswertes Ziel, dem sich aber sogar die meisten erfolgreichen Systeme widersetzen. Lehman's Law der Softwareevolution, und insbesondere das kontinuierliche Wachstum, beschreibt den ständigen Druck, neue Funktionalitäten hinzuzufügen, wenn Systeme genutzt und neue Anforderungen oder Möglichkeiten wahrgenommen werden. Die Fähigkeit, mit dieser zunehmenden Komplexität nicht nur irgendwie umzugehen, sondern sie sogar gewinnbringend zu nutzen, erhöht die Bedeutung von zwei Bereichen des Designs: dem Design von Systemen und dem Design der Organisation, die Systeme erstellt und weiterentwickelt. Wir verfügen über eine beträchtliche Anzahl von Veröffentlichungen, die sich auf den ersten Bereich konzentrieren, d. h. auf System- und Softwaredesign und -architektur, einschließlich einer ständig wachsenden Anzahl von Büchern über Domain-driven Design und Softwarearchitektur. Team Topologies befasst sich mit der Gestaltung der Softwareentwicklungsorganisation unter Berücksichtigung von Conway's Law.

Die Grundthese [...] ist, dass Organisationen, die Systeme entwerfen [...], gehalten sind, Entwürfe zu erstellen, die eine Kopie der Kommunikationsstrukturen dieser Organisationen sind. Wir haben gesehen, dass diese Tatsache wichtige Implikationen für das Management des System-Designs hat. In erster Linie haben wir ein Kriterium für die Strukturierung von Design-Organisationen gefunden: Ein Design-Aufwand sollte nach dem Bedarf an Kommunikation organisiert werden.¹

Das obige Zitat aus der Schlussfolgerung von Mel Conways klassischem Paper über Organisationsdesign für Softwareentwicklung ist ein sehr passender Anfang für dieses Buch. Team Topologies beschreibt organisatorische Muster für die Teamstruktur und die Art der Interaktion, wobei die Kraft, die die Organisation auf das System ausübt, als ein treibendes Design-Anliegen betrachtet wird.

Mit zunehmender Komplexität des Systems steigen in der Regel auch die kognitiven Anforderungen an die Organisation, die das System aufbaut und weiterentwickelt. Die Beherrschung der kognitiven Belastung durch Teams mit klaren Zuständigkeiten und Grenzen ist ein besonderer Schwerpunkt der Teamgestaltung im

1 Conway, »How Do Committees Invent?«

Rahmen des Team-Topologies-Ansatzes. Zur Erzielung angemessener, abgegrenzter Verantwortlichkeiten wird eine natürliche und relativ unabhängige System-(Unter-)Struktur angestrebt, an der sich die Teams dann ausrichten. Dabei wird das Conway'sche Gesetz berücksichtigt und genutzt, um kohäsive Strukturen mit klaren Grenzen und loser Kopplung zu gewährleisten (bekannt als umgekehrtes Conway-Manöver, das hier beschrieben wird).

Wenn das alles wäre, wäre *Team Topologies* eine nützliche Ausarbeitung von Conways Paper und würde sie in den aktuellen Kontext stellen. Natürlich ist Team Topologies noch mehr als das. Insbesondere werden vier Team-Patterns aufgezeigt und ihre Ergebnisse, ihre Form und die Kräfte, die sie betreffen und von denen sie geprägt werden, beschrieben. Stream-aligned Teams sind die primäre Teamform. Diese Teams sind für den Arbeitsfluss optimiert und verfügen über alles, was sie brauchen, um eine kontinuierliche Wertschöpfung zu erzielen und auf die damit verbundenen Rückkopplungszyklen voll zu reagieren. Das bedeutet, dass beim System-Design nicht nur eine lose Kopplung angestrebt wird, sondern eine Dekomposition, die den Arbeitsfluss unterstützt und die Abhängigkeiten und den Koordinationsbedarf zwischen Stream-aligned Teams verringert. Complicated-subsystem Teams und Platform Teams reduzieren die Belastung der Stream-aligned Teams, wobei letztere interne Kunden der Subsystem- oder Plattformfähigkeiten der ersteren sind (und alle Phasen der Entwicklung, Lieferung und des Betriebs für mehrere Stream-Teams unterstützen). Enabling Teams dienen ebenfalls anderen Teams, aber sie sind Dienstleister, die den Stream-aligned Teams dabei helfen, neue Techniken zu erlernen, neue Technologien zu erforschen usw., sodass die Stream-aligned Teams ihren Fokus beibehalten und gleichzeitig ihre Effektivität steigern können.

Matthew Skelton und Manuel Pais haben ihre beträchtliche Erfahrung eingebracht, indem sie beschrieben haben, was diese verschiedenen Teamformen brauchen, um erfolgreich zu sein. Sie bringen aber auch Variationen im Kontext hervor, identifizieren die daraus resultierenden Designimplikationen und weisen auf zu vermeidende Anti-Patterns hin. Mit enormer Großzügigkeit fügen sie auch Erkenntnisse aus einschlägigen Arbeiten hinzu und weisen auf diese hin. Zusammen mit einer Reihe von Fallstudien ergänzt dies das Buch weiter.

Team Topologies verbessert und bereichert unser Verständnis der Organisationsarchitektur durch die nuancierte Darstellung der wichtigsten Strukturmuster, Interaktionsmodi oder Dynamiken und Überlegungen zur Evolution. Aufgrund seiner Klarheit und Fokussierung dient es als pragmatischer Leitfaden für die Bildung von Teams und die Bewältigung ihrer Herausforderungen oder für die Unterstützung bestehender Teams bei der effektiven Wertschöpfung.

– **Ruth Malan**, Architektur-Consultant bei
Bredemeyer Consulting

Vorwort des Übersetzers

Als die englische Version von Team Topologies im Jahr 2019 erschien, wurde das Buch sehr schnell durch viele von mir geschätzte Fachleute, insbesondere aus der Domain-driven Design Community, erwähnt. Es dauerte auch nicht lange, bis Team Topologies Einzug in meinen Alltag als Berater im Umfeld soziotechnischer Architekturen hielt. Viele meiner Kunden wollten mit modernen technischen Architekturen und kollaborativer fachlicher Modellierung ein schnelleres Time-to-Market erreichen. Team Topologies füllte an dieser Stelle eine wichtige Lücke, indem es eine gut kommunizierbare und verständliche Idee sowie eine Terminologie für Teams und ihre Interaktionsformen lieferte.

Team Topologies zu übersetzen, war für mich eine spannende Herausforderung. Es war selbstverständlich mein Ziel, dass die Essenz und die Inhalte in dieser deutschen Ausgabe klar und verständlich vermittelt werden. Weil zentrale englische Begriffe der Team-Topologies-Terminologie wie die vier grundlegenden Teamtypen und drei Interaktionsmodi auch in der deutschsprachigen Community verwendet werden, habe ich sie in meiner Übersetzung beibehalten.

Diese Übersetzung wäre ohne die Unterstützung einiger Personen nicht möglich gewesen. Mein aufrichtiger Dank geht an zahlreiche Expertinnen und Experten aus meinem Umfeld, die mich seit vielen Jahren mit ihrem Fachwissen und ihrer Hingabe unterstützt und inspiriert haben. Im Kontext dieser Übersetzung ist es mir ein Anliegen insbesondere Jochen Christ, Anja Kammer, Simon Harrer, Gerrit Beine, Susanne Kaiser und Henning Schwentner namentlich zu erwähnen. Ein besonderer Dank gilt auch den Autoren Matthew Skelton und Manuel Pais, die mit ihrem Buch eine unverzichtbare Ressource für die Gestaltung moderner Delivery-Organisationen geschaffen haben und in zahlreichen persönlichen Gesprächen auf Konferenzen immer spannende Impulsgeber für mich sind. Mein Dank gilt auch Ariane Hesse von O'Reilly Deutschland für die stets angenehme und produktive Zusammenarbeit. Abschließend möchte ich auch meiner Frau und meinen beiden Stieftöchtern für ihre Geduld und ihre Unterstützung während der Übersetzungsarbeiten danken.

Widmen möchte ich diese Übersetzung dem viel zu früh verstorbenen Stefan Tilkov, der für mich weit mehr als ein Vorgesetzter war. Stefan war für mich immer ein wichtiger Mentor, ein fordernder Sparringspartner und vor allem ein Vorbild. Ruhe in Frieden Stefan.

Ich hoffe, dass Ihnen diese Übersetzung ebenso viel Einsichten und Inspiration bietet wie mir vor einigen Jahren das Original.

– **Michael Plöd**
Fellow @ INNOQ

Einleitung

Bei [modernem] Organisationsdesign geht es um die Gestaltung für kollaborative Technologien, für die Stimme der Kundschaft.

– Naomi Stanford, *Guide to Organization Design*

Teams befinden sich ständig im Wandel, aber sie sind auch Ihre beste Chance, kontinuierlich und nachhaltig Wertschöpfung zu erzielen, indem Sie sie auf das Geschäft ausrichten. Idealerweise sollten Teams langlebig und autonom sein und über engagierte Teammitglieder verfügen. Teams leben jedoch nicht in Isolation. Sie müssen verstehen, wie und wann sie miteinander interagieren. Und diese Teaminteraktionen müssen sich im Laufe der Zeit weiterentwickeln, um die verschiedenen Phasen der Entwicklung und Umsetzung zu unterstützen, die Produkte und Technologien während ihrer Lebensdauer durchlaufen. Kurz gesagt: Unternehmen müssen nicht nur autonome Teams anstreben, sondern auch sich selbst ständig überdenken und weiterentwickeln, um den Kundinnen und Kunden schnell einen Mehrwert zu bieten.

Dieses Buch bietet ein praktisches, schrittweises, anpassungsfähiges Modell für die Gestaltung von Organisationen, das wir in Unternehmen mit unterschiedlichem Reifegrad eingesetzt haben und das sich bewährt hat: Team Topologies.

Team Topologies ist jedoch keine allgemeingültige Formel, um Softwaresysteme erfolgreich zu entwickeln und zu betreiben. Es gibt Teams und Organisationen, die mit einer ganz anderen Organisationsdynamik als der hier beschriebenen und empfohlenen erfolgreich sind (insbesondere in Organisationen mit einer ausgezeichneten Kultur und bereits vorhandenen Best Practices).

Team Topologies soll klare Muster liefern, die für viele verschiedene Teams und Organisationen einfach zu befolgen und zu interpretieren sind, und nicht herausragenden Spielern vorschreiben, wie sie zu spielen haben. Wir betrachten Team Topologies als eine Reihe von Musikstücken für ein Orchester oder eine Big Band, nicht als die Melodie für eine Top-Jazztrompeterin. Gedruckte Noten für ein großes Musikensemble tragen zum Erfolg der Gruppe bei, schreiben aber nicht jeden Aspekt der Darbietung vor. Viele Details bleiben dem Ensemble überlassen, um sie

je nach Anlass, Veranstaltungsort oder Zusammensetzung der Spielerinnen und Spieler zu interpretieren. Ebenso ist es von großem Wert, sich auf ein kohärentes Vokabular und eine gemeinsame Arbeitsweise in den Teams zu einigen, um eine gute Bereitstellung von Software zu erreichen.

Der Ansatz von Team Topologies hilft Organisationen, die Schwierigkeiten haben, einen Weg zur Optimierung ihrer Teamstruktur zu finden, oder denen noch nicht bewusst ist, welche Auswirkungen die Gestaltung von Teams auf gute Geschäftsergebnisse und insbesondere Softwaresysteme haben kann. Team Topologies hilft Unternehmen, schneller und kontinuierlicher als bisher erfolgreich zu sein.

Dieses Buch richtet sich an alle, denen die Effizienz der Bereitstellung und des Betriebs von Softwaresystemen am Herzen liegt: Führungskräfte auf C-Level (einschließlich CTOs/CIOs, CEOs, CFOs usw.), Manager, Abteilungsleiterinnen, Softwarearchitekten und Systemarchitektinnen sowie alle anderen, die an der Entwicklung oder dem Betrieb von Softwaresystemen beteiligt sind und die Bereitstellung und den Betrieb dieser Systeme effektiver gestalten wollen oder müssen.

Wie wir dazu kamen, dieses Buch zu schreiben

Im Jahr 2013 entwickelte Matthew in einem Blogbeitrag mit dem Titel »What Team Structure Is Right for DevOps to Flourish?«¹ die ursprünglichen DevOps-Topologien (und Anti-Patterns), während er bei einem Unternehmen in Großbritannien DevOps und Continuous Delivery einführte. Damals kämpfte das Unternehmen, das er beriet, mit der Einführung moderner Ansätze für die Bereitstellung von Software, und die frühen Topologiemuster, die Matthew entwickelte, boten dem Unternehmen eine Möglichkeit, verschiedene Optionen zu erkunden.

Manuel interviewte Matthew auf der Softwareentwicklungskonferenz QCon London im Jahr 2015, wo Matthew über das Conway'sche Gesetz und die frühen DevOps-Topology-Muster sprach. Der daraus resultierende Artikel »How Different Team Topologies Influence DevOps Culture« (Wie verschiedene Team-Topologien die DevOps-Kultur beeinflussen) wurde von InfoQ veröffentlicht und in mehrere Sprachen übersetzt². Später im selben Jahr half Manuel dabei, die DevOps-Topology-Muster zu erweitern, und es gab hierzu Beiträge aus der Community.

Seitdem hat sich die Verwendung der DevOps-Topology-Muster sprunghaft erhöht. In Vorträgen, Artikeln und Gesprächen wurde immer wieder auf sie Bezug genommen. Sie haben Organisationen aller Größen und Branchen auf der ganzen Welt dabei geholfen, über die Beziehungen zwischen Teams nachzudenken und darüber, wie ihre Interaktionen sowohl die Organisationskultur als auch die Softwarearchitektur beeinflussen.

1 Skelton, »What Team Structure Is Right for DevOps to Flourish?«.

2 Skelton, »How Different Team Topologies Influence DevOps Culture«.

Im Laufe der Zeit wurde uns klar, dass die ursprünglichen DevOps-Topologien eine statische Sicht auf die Beziehungen zwischen den Teams darstellten, die zwar für die ersten Diskussionen nützlich, aber in ihrem Umfang recht begrenzt war. Durch unsere Erfahrung mit Schulungs- und Beratungsunternehmen aus der ganzen Welt haben wir festgestellt, dass einige Teams besser relativ isoliert oder autonom arbeiten, während andere Teams besser mit starker Kollaboration arbeiten. Wir haben uns gefragt, warum das so ist, und wir haben unsere Ideen auf der Grundlage des Feedbacks unserer Kunden weiterentwickelt.

Dies führte letztendlich zu den Team Topologies, wie sie in diesem Buch vorgestellt werden: ein dynamischer und sich weiterentwickelnder Ansatz zur Gestaltung von Organisationen, der auf realen Szenarien aus verschiedenen Regionen und Branchen basiert.

Wie man dieses Buch benutzt

Team Topologies soll ein praktisches Buch sein. Wir haben die Absicht, Inhalte zu vermitteln, die interaktiv sind und so viel Lernstoff liefern, wie wir auf diesen Seiten unterbringen können. Deshalb haben wir einige Design-Entscheidungen getroffen, die Ihnen die Navigation in diesem Buch erleichtern werden.

Zum einen ist das *Team Topologies* in drei Teile gegliedert:

Teil I des Buchs untersucht das Conway'sche Gesetz, die Art und Weise, wie organisatorische Zusammenhänge die Gestaltung der Systeme, die wir schaffen, einschränken, und wie wir diese Neigung zu unserem Vorteil nutzen können. Anschließend definieren wir, was wir unter Teams verstehen, und betrachten einige praktische Einschränkungen, die eine effektive Teamarbeit beeinflussen.

In Teil II untersuchen wir eine Reihe von statischen Team-Patterns, die sich in der Branche bewährt haben, und die Auswirkungen bei der Wahl eines Patterns gegenüber einem anderen unter Berücksichtigung des Conway'schen Gesetzes und des organisatorischen Kontexts. Dieser Teil soll Ihnen dabei helfen, über Team-Topologien nachzudenken, die im Großen und Ganzen für Ihren organisatorischen Kontext geeignet sind. Dieser Teil gibt Ihnen auch eine Hilfestellung bei der Entscheidung, wie die Teams auf die einzelnen Komponenten eines Systems ausgerichtet werden können, wobei das Conway'sche Gesetz und die grundlegenden Team-Topologien berücksichtigt werden.

In Teil III schließlich befassen wir uns mit der Frage, wie sich die Organisation weiterentwickeln lässt, um leistungsstarke Fähigkeiten für Innovation und schnelle Lieferung als Reaktion auf einen sich schnell ändernden operativen Kontext bereitzustellen. Wir erläutern, wie Sie den Team-Topologies-Ansatz nutzen können, um eine sensibilisierte Organisation zu schaffen, die auf die Anforderungen des Markts und der Nutzerinnen reagiert und die Auswirkungen auf die Einstellung von Mitarbeitern und deren Fähigkeiten berücksichtigt.

Jeder Teil beginnt mit einer Aufschlüsselung der wichtigsten Aussagen aus den einzelnen Kapiteln. Überall in den Kapiteln haben wir Abbildungen und Hinweise eingefügt, um Informationen hervorzuheben, die unserer Meinung nach hilfreich sind. Außerdem stellen wir Ihnen leicht nachvollziehbare Szenarien, Fallstudien und konkrete Empfehlungen für verschiedene Situationen zur Verfügung.

Und schließlich haben auch die Formen, Farben und Muster, die in vielen der Figuren zu finden sind, eine durchgängige Bedeutung im gesamten Buch. Hier ist die Legende:

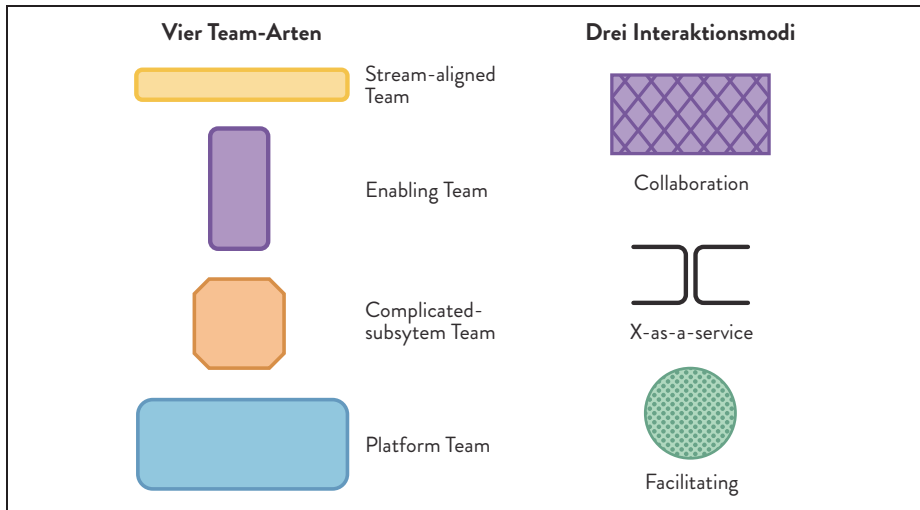


Abbildung 1: **Die vier Teamtypen und drei Interaktionsmodi**

Für ein umfassendes Verständnis sollten Sie das Buch von der ersten bis zur letzten Seite lesen, denn der Stoff baut sich Kapitel für Kapitel auf. Wir haben das Material jedoch so geschrieben, dass jeder Abschnitt relativ unabhängig ist.

In diesem Sinne finden Sie hier einige Szenarien mit entsprechenden Möglichkeiten, das Buch zu lesen, die zu Ihrer aktuellen Situation passen könnten:

- Ich brauche mehr Klarheit über die verschiedenen Teamtypen und welche Teamtypen effektiv sind.
 - Lesen Sie Kapitel 1 (Überblick), dann Kapitel 4 (Statische Topologien), dann Kapitel 5 (Grundlegende Topologien).
- Ich muss ein großes, monolithisches Softwaresystem aufteilen.
 - Lesen Sie Kapitel 6 (Grenzen) und dann Kapitel 3 (Das Team).
- Ich muss die Architektur eines Softwaresystems verbessern.
 - Lesen Sie Kapitel 2 (Conways Gesetz), dann Kapitel 4 (Statische Topologien) und dann Kapitel 6 (Grenzen).

- Ich muss die Effektivität von Softwareentwicklungsteams verbessern.
 - Lesen Sie Kapitel 3 (Das Team), dann Kapitel 6 (Grenzen), dann Kapitel 5 (Grundlegende Topologien).
- Ich muss die Moral und Effektivität innerhalb der Teams verbessern.
 - Lesen Sie Kapitel 3 (Das Team) und dann Kapitel 5 (Grundlegende Topologien).
- Ich muss verstehen, wo ich Aufwand betreiben muss, um das erwartete Wachstum zu erreichen.
 - Lesen Sie Kapitel 1 (Überblick), dann Kapitel 5 (Grundlagen), dann Kapitel 8 (Topologie-Evolution).
- Ich muss verstehen, wie ich Team-Topologien weiterentwickeln kann, um sich ändernden Business-Anforderungen gerecht zu werden.
 - Lesen Sie Kapitel 7 (Dynamische Aspekte) und dann Kapitel 8 (Topologie-Evolution und organisatorisches Erkennen).

Wichtige Einflüsse, die dieses Buch geprägt haben

Neben unseren eigenen Erfahrungen ist dieses Buch stark von mehreren verwandten Ansätzen und Denkweisen beeinflusst. Erstens gehen wir davon aus, dass eine Organisation ein sozio-technisches System oder Ökosystem ist, das durch die Interaktion von Individuen und Teams innerhalb der Organisation geformt wird; mit anderen Worten, dass eine Organisation die Interaktion zwischen Menschen und Technologie ist. In dieser Hinsicht deckt sich das Buch mit Ideen aus folgenden Bereichen: Kybernetik (insbesondere die Verwendung der Organisation als »Sensing-Mechanismus«, die bis ins Jahr 1948 zurückreicht, als Norbert Wiener sein Buch *Cybernetics: Or Control and Communication in the Animal and the Machine* veröffentlichte), Systems Thinking (insbesondere die Arbeit von W. Edwards Deming) und Ansätze wie das Cynefin-Framework zur Bewertung der Komplexität von Domänen (beschrieben von Dave Snowden und Mary Boone in ihrem 2007 in der *Harvard Business Review* erschienenen Artikel mit dem Titel »A Leader's Framework for Decision Making«) sowie die Theorie der adaptiven Strukturierung (ein Begriff, der von Gerardine DeSanctis und Marshall Scott Poole in ihrem *Organization Science*-Artikel »Capturing the Complexity in Advanced Technology Use« geprägt wurde: »Adaptive Structuration Theory«, in dem sie betonten, dass die Auswirkungen von Technologie keine Selbstverständlichkeit sind, sondern davon abhängen, wie Gruppen und Organisationen sie wahrnehmen).

Zweitens gehen wir davon aus, dass »das Team« etwas ist, das sich anders verhält als eine bloße Ansammlung von Individuen, und dass das Team in seiner Entwicklung und Funktionsweise gefördert und unterstützt werden sollte. In dieser Hinsicht stützen wir uns auf die Ideen von Bruce Tuckman (der in seinem Papier *De-*

developmental Sequence in Small Groups aus dem Jahr 1965 das Vier-Stufen-Modell – Forming, Storming, Norming, Performing – für die Entwicklung von Teams vorschlug), Russ Forrester und Allan Drexler (die in ihrem Paper *A Model for Team-Based Organization Performance* aus dem Jahr 1999 die Leistung teambasierter Organisationen untersuchten), Pamela Knight (die in ihrem Paper *Acquisition Community Team Dynamics: The Tuckman Model vs. the DAU Model* aus dem Jahr 2007 Beweise dafür fand, dass Storming während der gesamten Lebensdauer eines Teams stattfindet), Patrick Lencioni (der sich in seinem bahnbrechenden Buch *The Five Dysfunctions of a Team: A Leadership Fable* mit Fragen der Teaminteraktion beschäftigt) und ähnliche Theorien und Forschungen, die sich mit Teams befassen.

Drittens gehen wir davon aus, dass das Conway'sche Gesetz (oder eine Variante davon) ein starker Treiber für die Gestaltung von Softwareprodukten ist und dass Unternehmen davon profitieren würden, wenn sie sich explizit mit den Implikationen dieses Gesetzes auseinandersetzen würden. In dieser Hinsicht stützen wir uns auf Schriften und Ideen von Mel Conway, Ideen der Softwarearchitekturberaterin und Preisträgerin des Teamorganisations-Designs Ruth Malan, von James Lewis, dem technischen Direktor von ThoughtWorks und einem der Befürworter des »umgekehrten Conway-Manövers« sowie von ähnlichen Autoren und Praktikern.

Schließlich stützen wir uns auf zahlreiche Quellen, die praktische Erfolge bei der Entwicklung und dem Betrieb von Softwaresystemen in großem Maßstab dokumentieren, darunter Organisationen wie Adidas, Auto Trader, Ericsson, Netflix, Spotify, TransUnion und andere. Die Größe und Schnelligkeit dieser Unternehmen hat es ihnen ermöglicht, innerhalb von einigen Monaten bis hin zu einigen Jahren handfeste Vorteile aus Veränderungen in der Organisationsstruktur und der Interaktion zwischen den Teams zu erzielen.

Wir hoffen, dass Sie auf Ihrer Reise durch dieses Buch dazu inspiriert werden, Ihre Vorstellungen von Teams, ihren Strukturen und ihrer Funktionsweise zu hinterfragen.

TEIL I

**Teams als entscheidendes
Mittel bei der Lieferung**

Die wichtigsten Erkenntnisse

Kapitel 1

- Das Conway'sche Gesetz legt nahe, dass die gemeinsame Entwicklung von Softwarearchitekturen und Teaminteraktionen von großem Nutzen ist, da es sich um ähnliche Antriebskräfte handelt.
- Team Topologies klärt den Zweck und die Verantwortlichkeiten von Teams und steigert so die Effektivität ihrer Beziehungen untereinander.
- Team Topologies verfolgt einen humanistischen Ansatz bei der Entwicklung von Softwaresystemen und stellt gleichzeitig Organisationen für strategische Anpassungsfähigkeit auf.

Kapitel 2

- Unternehmen sind dazu gezwungen, Designs zu erstellen, die ihre Kommunikationswege widerspiegeln.
- Das Design der Organisation schränkt den »Suchraum nach Lösungen« ein und begrenzt die Möglichkeiten im Design von Software.
- Die Forderung, dass jede mit jedem kommunizieren muss, ist ein Rezept für Chaos.
- Wählen Sie Softwarearchitekturen, die einen teamorientierten Arbeitsfluss fördern.
- Die Beschränkung der Kommunikationswege auf genau definierte Teaminteraktionen führt zu modularen, entkoppelten Systemen.

Kapitel 3

- Das Team ist das effektivste Mittel zur Bereitstellung von Software, nicht einzelne Personen.
- Begrenzen Sie die Größe von Multi-Team-Gruppierungen innerhalb der Organisation auf der Grundlage von Dunbars Zahl.
- Schränken Sie die Zuständigkeiten der Teams so ein, dass sie der maximalen kognitiven Belastung des Teams entsprechen.
- Legen Sie klare Verantwortungsgrenzen für Teams fest.
- Ändern Sie das Arbeitsumfeld für Teams, um ihnen zum Erfolg zu verhelfen.

1

Das Problem mit Organisationsdiagrammen

Organisationen sollten als komplexe und anpassungsfähige Organismen und nicht als mechanistische und lineare Systeme betrachtet werden.

– Naomi Stanford, *Guide to Organisation Design*

Mitarbeiterinnen und Mitarbeiter im Technologiebereich sind ständig in Aktion: Sie erstellen und aktualisieren Systeme in einem unglaublichen Tempo und kombinieren verschiedene Arten von Technologien, um ein überzeugendes Benutzererlebnis zu schaffen. Mobile Anwendungen, Cloud-basierte Dienste, Webanwendungen und eingebettete, tragbare oder industrielle IoT-Geräte müssen alle effektiv zusammenarbeiten, um die gewünschten Geschäftsergebnisse zu erzielen.

Heutzutage beeinflussen diese Systeme fast jeden Aspekt unseres täglichen Lebens in einer Weise, die immer tiefgreifender ist. Wenn Software schlecht konzipiert ist – oder noch wichtiger, wenn die Interaktion zwischen der Software, dem Anbieter und den Kunden nicht übereinstimmt –, hat das negative Auswirkungen auf die Menschen. Sie können weit weg von zu Hause gestrandet sein, wenn eine Anwendung für Taxidienste ausfällt. Sie können möglicherweise ihre Miete nicht mehr bezahlen, wenn die Software oder die Prozesse für das Internetbanking ausfallen. Sie können sogar ihr Leben in Gefahr sehen, wenn ein medizinisches Gerät ausfällt. Noch nie war explizites sozio-technisches Design so wichtig wie heute.

Entwicklung und Betrieb dieser hochkomplexen, miteinander vernetzten Softwaresysteme sind Teamaktivitäten, die die gemeinsamen Anstrengungen von Menschen mit unterschiedlichen Fähigkeiten über verschiedene Plattformen hinweg erfordern. Darüber hinaus müssen moderne IT-Organisationen Softwaresysteme schnell und sicher bereitstellen und betreiben, während sie gleichzeitig wachsen und sich an Veränderungen und Zwänge im geschäftlichen oder gesetzlichen Umfeld anpassen müssen. Unternehmen können nicht mehr zwischen einer Optimierung der Stabilität und einer Optimierung der Geschwindigkeit wählen.

Doch trotz dieser Risiken und Anforderungen organisieren viele Unternehmen ihre Mitarbeiter und Teams immer noch auf eine Weise, die für die moderne Softwareentwicklung und den Betrieb kontraproduktiv ist. Unternehmen, die sich zu

sehr auf Organigramme und Matrixstrukturen verlassen, um die Arbeit aufzuteilen und zu kontrollieren, schaffen oft nicht die notwendigen Voraussetzungen, um Innovationen zu fördern und gleichzeitig ein hohes Arbeitstempo zu erreichen. Um dies zu realisieren, brauchen Unternehmen stabile Teams, effektive Teamstrukturen und -interaktionen. Sie müssen in fähige, kompetente Teams investieren, die die Grundlage für Agilität und Anpassungsfähigkeit bilden. Um in einem immer härter umkämpften Markt bestehen zu können, brauchen Unternehmen Teams und Mitarbeiterinnen und Mitarbeiter, die in der Lage sind, zu erkennen, wann sich das Umfeld ändert, um sich entsprechend weiterzuentwickeln.

Die gute Nachricht ist, dass es möglich ist, gleichzeitig schnell und sicher zu sein – mit der richtigen Einstellung und mit Tools, die sowohl Anpassungsfähigkeit als auch Wiederholbarkeit fördern und die Teams und Menschen in den Mittelpunkt stellen. Wie Mark Schwartz und seine Mitautoren in ihrem 2016 erschienenen Buch *Thinking Environments* schreiben, »muss die Organisationsstruktur die Verantwortlichkeiten so koordinieren, dass das Ziel, qualitativ hochwertige und wirkungsvolle Software zu liefern, unterstützt wird.«¹

Als Mitglieder der Technologieteams, die diese Schnittstellen betreuen, müssen wir unser Denken ändern. Wir müssen Teams nicht mehr als Ansammlung austauschbarer Individuen betrachten, die erfolgreich sind, solange sie den »richtigen« Prozess befolgen und die »richtigen« Werkzeuge verwenden, sondern wir müssen Menschen und Technologie als ein einziges sozio-technisches Mensch-Computer/Kohlenstoff-Silizium-Ökosystem betrachten. Gleichzeitig müssen wir sicherstellen, dass die Teams intrinsisch motiviert sind und eine echte Chance haben, in einem solchen Gefüge ihre beste Arbeit zu leisten.

In diesem Kapitel wird Team Topologies als ein anpassungsfähiges Modell für die Gestaltung von Technologieorganisationen vorgestellt, mit dem Unternehmen Geschwindigkeit und Stabilität gleichermaßen erreichen können. Doch zunächst wollen wir uns ansehen, wie sich die realen Kommunikationsstrukturen in den meisten Organisationen oft von dem unterscheiden, was uns das Organigramm verrät, und welche Auswirkungen das hat.

Kommunikationsstrukturen einer Organisation

Die meisten Unternehmen möchten oder müssen eine genormte Sicht auf ihre Teams und Mitarbeiter haben, das sogenannte »Organigramm«. In diesem Diagramm werden die Teams, Abteilungen, Referate und andere Organisationseinheiten sowie ihre Beziehungen zueinander dargestellt. Es zeigt in der Regel hierarchische Berichtslinien, die Kommunikationswege »von oben nach unten« in der Organisation implizieren.

1 Schwartz et al., »Thinking Environments«, 21.

Das Organigramm hat im Zusammenhang mit der Entwicklung von Softwaresystemen durchaus seine Berechtigung, insbesondere im Hinblick auf die Einhaltung von Vorschriften und gesetzlichen Bestimmungen. In einem hochgradig kollaborativen Kontext mit Ungewissheit über die Resultate führt es jedoch zu unrealistischen Erwartungshaltungen, wenn man sich auf das Organigramm als zentralen Mechanismus für das Aufteilen der zu erledigenden Arbeit verlässt. Wir müssen uns stattdessen auf entkoppelte, langlebige Teams verlassen, die effektiv zusammenarbeiten können, um die Herausforderung zu meistern, ein Gleichgewicht zwischen Geschwindigkeit und Verlässlichkeit herzustellen.

Das Problem, wenn wir das Organigramm für bare Münze nehmen, besteht darin, dass wir versuchen, Menschen so zu strukturieren, als wären sie eine Software, die ihre Kommunikation ordentlich innerhalb der vorgegebenen Leitlinien hält. Aber die Menschen beschränken ihre Kommunikation nicht nur auf die Linien im Organigramm. Wir wenden uns an diejenigen, auf die wir angewiesen sind, um unsere Arbeit zu erledigen. Wir weichen die Regeln auf, wenn es nötig ist, um unsere Ziele zu erreichen. Deshalb sehen die tatsächlichen Kommunikationslinien ganz anders aus als im Organigramm, wie in Abbildung 1-1 dargestellt.

Das Denken in Organigrammen ist das Problem

Wie in Abbildung 1-1 dargestellt, helfen uns herkömmliche Organigramme nicht beim Verständnis der tatsächlichen Kommunikationsmuster in unserer Organisation. Stattdessen müssen Unternehmen ein realistischeres Bild von der erwarteten und der tatsächlich stattfindenden Kommunikation zwischen Einzelpersonen und Teams entwickeln. Diese Diskrepanzen geben Aufschluss darüber, welche Arten von Systemen für das Unternehmen besser geeignet sind.

Darüber hinaus neigen Entscheidungen, die auf der Struktur von Organigrammen basieren, dazu, nur für einen Teil des Unternehmens zu optimieren und die vor- und nachgelagerten Auswirkungen zu ignorieren. Lokale Optimierungen helfen den direkt involvierten Teams, aber sie tragen nicht unbedingt dazu bei, die Wertschöpfung für die Kunden insgesamt zu verbessern. Ihre Auswirkungen sind möglicherweise vernachlässigbar, wenn es größere Engpässe im Arbeitsablauf gibt. Wenn beispielsweise Teams Cloud as Code und Infrastructure as Code einsetzen, kann sich die Zeit für die Bereitstellung einer neuen Infrastruktur von Wochen oder Monaten auf Minuten oder Stunden verkürzen. Wenn jedoch jede Änderung von einem Gremium, das sich einmal pro Woche trifft, genehmigt werden muss, dann wird die Geschwindigkeit der Bereitstellung bestenfalls wöchentlich sein.

Systems Thinking konzentriert sich darauf, ganzheitlich zu optimieren, den gesamten Arbeitsfluss zu betrachten, festzustellen, was heute der größte Engpass ist, und ihn zu beseitigen. Dann wiederholen Sie das. Team Topologies konzentriert sich darauf, wie man dynamische Teamstrukturen und Interaktionsmodi einrichtet, die den Teams helfen, sich schnell an neue Bedingungen anzupassen und eine schnelle und sichere Softwareauslieferung zu erreichen. Vielleicht ist das heute noch nicht