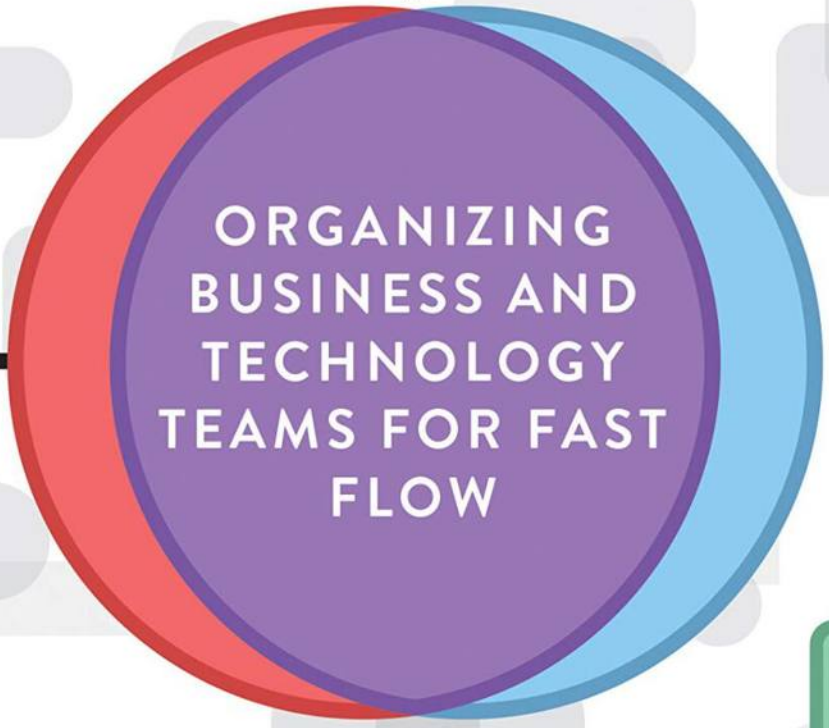
TEAM TOPOLOGIES



ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW

Foreword by RUTH MALAN

MATTHEW SKELTON and MANUEL PAIS

Praise for
TEAM TOPOLOGIES

“Team Topologies provides fresh insights on how to anticipate and adapt to market and technology changes. To survive, enterprises need to unlearn existing command and control structures and instead move authority to leaders with the best information to take action and respond. This book will help executives and business leaders focus on the key strategies of high-performance teams to effectively address the needs of today and the evolving landscape of tomorrow.”

—Barry O’Reilly, Founder of ExecCamp, Business Advisor, and Author of Unlearn and Lean Enterprise

“There is nothing more fundamental to management than how you structure your organization and what behaviors you encourage. Despite this, few have attempted to catalog and analyze the organizational design patterns of IT organizations going through digital, DevOps, and SRE transformations. Skelton and Pais have not only accepted this bold challenge, but they’ve also hit the mark by creating an indispensable and unique resource.”

—Damon Edwards, Co-Founder of Rundeck

“Team Topologies provides a much-needed framework for evaluating and optimizing team organization for increased flow. Teams that have the right size, the right boundaries, and the right level of communication are poised to deliver value to the company and satisfaction to the team members. Team Topologies combines a methodical approach with real-world case studies to unlock the full potential of your tech teams.”

—Greg Burrell, Senior Reliability Engineer at Netflix

“Team Topologies by Matthew Skelton and Manuel Pais is unique. It is going to have a big influence across tech companies. We need a structured and methodical approach to shaping teams for continuous delivery instead of copying a few Spotify rituals. This is the book.”

—Nick Tune, API Platform Lead, Navico

“At Condé Nast International, [the DevOps Topologies] was crucial in understanding our current DevOps state and in defining the vision for our aspirational DevOps operating model. We were able to navigate round the pitfalls and organizational anti-patterns as excellently described in the models. . . . I am extremely pleased that Matthew and Manuel are growing on the success of the DevOps Topologies and turning their further learnings into the far-reaching book Team Topologies for organization design.”

—Crystal Hirschorn, VP of Engineering, Global Strategy and Operations at Condé Nast

“The high-performing team is the core generator of value in the modern digital economy. But cultivating and scaling an adaptive ecosystem of such teams is a too-often elusive goal. In Team Topologies, Skelton and Pais provide innovative tools and concepts for structuring the next generation digital operating model. Recommended for CIOs, enterprise architects, and digital product strategists worldwide.”

—Charles Betz, Principal Analyst, Forrester Research

“Matthew Skelton and Manuel Pais say ‘Team Topologies is meant to be a functional book’—and it is. It’s well constructed and sign-posted, based in sound thinking, and challenges readers to assume, like them, that an organization is a sociotechnical system or ecosystem. From this assumption comes practical suggestions, no prescriptions, and skill in explaining an approach that provides for effective tech/human organization design. For anyone in the tech/organization design field, [Team Topologies is] well worth reading.”

—Dr. Naomi Stanford, Organization Design Practitioner, Teacher, and Author

“I have found Matthew and Manuel’s work on patterns and language to be incredibly valuable in both shaping strategies to transform team contexts over time across our organization, as well as in helping business and technology leadership connect with the topics of flow and continuous delivery.”

—Richard James, Head of Digital Technology & Engineering at Nationwide

“Teams are the fundamental building block of organizations, how those teams work and the system they operate in are the difference between average and high performance. This book is a deep well of information for how you can optimize your organization’s system for your current context.”

—Jeremy Brown, Director, Red Hat Open Innovation Labs EMEA

“DevOps is great, but how do real-world organizations actually structure themselves to do it? You can’t just put everyone on a single, silo-less team, all sitting together in one giant open-plan office and going out to lunch or playing foosball together. Team Topologies provides a practical set of templates for addressing the key DevOps question that other guides leave as an exercise for the student.”

—Jeff Sussna, Founder & CEO, Sussna Associates, and Author of Designing Delivery

“If you’re looking for an analysis of the challenges with the traditional ways of working, and also some practical guidance on mitigation strategies (e.g., new interaction modes, reducing cognitive load, and creating appropriate ‘Team APIs’), then this is the book for you!”

—Daniel Bryant, Technical Consultant/Advisor and News Manager at InfoQ

“Team Topologies makes for a fascinating read as it explores the symbiotic relationship between teams and the IT architecture they support. It goes beyond the common approach of static org charts or self-organizing chaos and shows how to evolve the people system and IT system together.”

—Mirco Hering, Global DevOps Lead Accenture and Author of DevOps for the Modern Enterprise

TEAM TOPOLOGIES

ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW

MATTHEW SKELTON and MANUEL PAIS

Foreword by Ruth Malan

IT Revolution
Portland, Oregon

25 NW 23rd Pl, Suite 6314
Portland, OR 97210

Copyright © 2019 by Matthew Skelton and Manuel Pais
For information about permission to reproduce selections from this book, write to Permissions, IT Revolution Press, LLC, 25 NW 23rd Pl, Suite 6314, Portland, OR 97210.

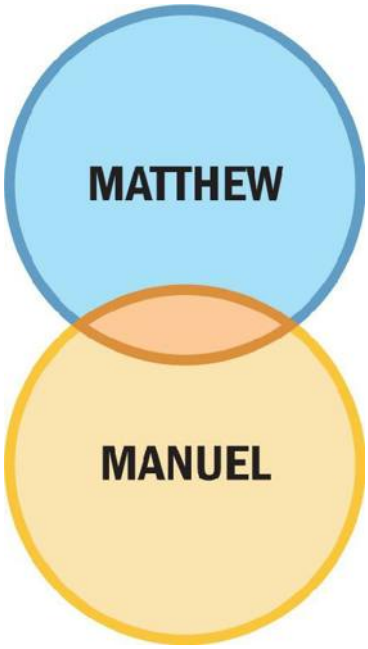
Cover and book design by Devon Smith

Library of Congress Catalog-in-Publication Data
Available upon request

ISBN: 978-1942788-812
eBook ISBN: 978-1942788-829
Kindle ISBN: 978-1942788-836
Web PDF ISBN: 978-1942788-843

For information about special discounts for bulk purchases, or for information on booking authors for an event, please visit our website atITRevolution.com.

TEAM TOPOLOGIES



To my wife, Suzy Beck—for all your support and inspiration.

To Katie, my life partner and family stronghold—thanks for your tireless love and support.

To Dan and Ben, daily sources of warmth—hopefully this book can help you understand what Daddy does for a living.

CONTENTS

Figures & Tables
Case Studies & Industry Examples
Foreword by Ruth Malan
Preface

PART I TEAMS AS THE MEANS OF DELIVERY

Chapter 1: The Problem with Org Charts
- Communication Structures of an Organization
- Team Topologies: A New Way of Thinking about Teams
- The Revival of Conway's Law
- Cognitive Load and Bottlenecks
- Summary: Rethink Team Structures, Purpose, and Interactions

Chapter 2: Conway's Law and Why It Matters
- Understanding and Using Conway's Law
- The Reverse Conway Maneuver
- Software Architectures that Encourage Team-Scoped Flow
- Organization Design Requires Technical Expertise
- Restrict Unnecessary Communication
- Beware: Naive Uses of Conway's Law
- Summary: Conway's Law Is Critical for Efficient Team Design in Tech

Chapter 3: Team-First Thinking
- Use Small, Long-Lived Teams as the Standard
- Good Boundaries Minimize Cognitive Load
- Design "Team APIs" and Facilitate Team Interactions
- Warning: Engineering Practices Are Foundational
- Summary: Limit Teams' Cognitive Load and Facilitate Team

Interactions to Go Faster

PART II TEAM TOPOLOGIES THAT WORK FOR FLOW

Chapter 4: Static Team Topologies
Team Anti-Patterns
Design for Flow of Change
DevOps and the DevOps Topologies
Successful Team Patterns
Considerations When Choosing a Topology
Use DevOps Topologies to Evolve the Organization
Summary: Adopt and Evolve Team Topologies that Match Your Current Context

Chapter 5: The Four Fundamental Team Topologies
Stream-Aligned Teams
Enabling Teams
Complicated-Subsystem Teams
Platform Teams
Avoid Team Silos in the Flow of Change
A Good Platform Is “Just Big Enough”
Convert Common Team Types to the Fundamental Team Topologies
Summary: Use Loosely Coupled, Modular Groups of Four Specific Team Types

Chapter 6: Choose Team-First Boundaries
A Team-First Approach to Software Responsibilities and Boundaries
Hidden Monoliths and Coupling
Software Boundaries or “Fracture Planes”
Real-World Example: Manufacturing
Summary: Choose Software Boundaries to Match Team Cognitive Load

PART III EVOLVING TEAM INTERACTIONS FOR INNOVATION AND RAPID DELIVERY

Chapter 7: Team Interaction Modes

- Well-Defined Interactions Are Key to Effective Teams
- The Three Essential Team Interaction Modes
- Team Behaviors for Each Interaction Mode
- Choosing Suitable Team Interaction Modes
- Choosing Basic Team Organization
- Choose Team Interaction Modes to Reduce Uncertainty and Enhance Flow
- Summary: Three Well-Defined Team Interaction Modes

Chapter 8: Evolve Team Structures with Organizational Sensing

- How Much Collaboration Is Right for Each Team Interaction?
- Accelerate Learning and Adoption of New Practices
- Constant Evolution of Team Topologies
- Combining Teams Topologies for Greater Effectiveness
- Triggers for Evolution of Team Topologies
- Self Steer Design and Development
- Summary: Evolving Team Topologies

Conclusion: The Next-Generation Digital Operating Model

- Four Team Types and Three Interaction Modes
- Team-First Thinking: Cognitive Load, Team API, Team-Sized Architecture
- Strategic Application of Conway’s Law
- Evolve Organization Design for Adaptability and Sensing
- Team Topologies Alone Are Not Sufficient for IT Effectiveness
- Next Steps: How to Get Started with Team Topologies

Glossary

Recommended Reading
References
Notes
Index
Acknowledgments
About the Authors

FIGURES & TABLES

FIGURES

0.1: The Four Team Types and Three Interaction Modes
1.1: Org Chart with Actual Lines of Communication
1.2: Obstacles to Fast Flow
2.1: Four Teams Working on a Software System
2.2: Software Architecture from Four-Team Organization
2.3: Microservices Architecture with Independent Services and Data Stores
2.4: Team Design for Microservices Architecture with Independent Services and Data Stores
2.5: Inter-Team Communication
3.1: Scaling Teams Using Dunbar's Number
3.2: No More than One Complicated or Complex Domain per Team
3.3: Typical vs. Team-First Software Subsystem Boundaries
3.4: Office Layout at CDL
4.1: Organization not Optimized for Flow of Change
4.2: Organization Optimized for Flow of Change
4.3: Relationship between SRE Team and Application Team
4.4: Influence of Size and Engineering Discipline on Team Interaction Patterns
5.1: The Four Fundamental Team Topologies
5.2: Platform Composed of Several Fundamental Team Topologies
5.3: Traditional Infrastructure Team Organization
5.4: Support Teams Aligned to Stream of Change
6.1: Mobile, Cloud, and IoT Technology Fracture Plane Scenario
7.1: Collaboration vs. X-as-a-Service
7.2: The Three Essential Team Interaction Modes
7.3: Team Interaction Modes Scenario

7.4: X-as-a-Service Team Interaction Mode
7.5: Primary Interaction Modes for the Four Fundamental Team Topologies
7.6: Team Interaction Modes at IBM around 2014
8.1: Collaboration between Cloud and Embedded Teams
8.2: System Build and Platform Build Team at TransUnion
8.3: System Build and Platform Build Team Collaboration at TransUnion
8.4: System Build and Platform Build Teams Merged at TransUnion
8.5: System Build and Platform Build Teams Merged Back Into Dev and Ops at TransUnion
8.6: Evolution of Team Topologies
8.7: Evolution of Team Topologies in an Enterprise
8.8: Example of a “Platform Wrapper”
8.9: New-Service and “Business as Usual” (BAU) Teams
8.10: Side-by-Side New Service and BAU Teams
9.1: Core Ideas of Team Topologies

TABLES

Table 7.1: Advantages and Disadvantages of Collaboration Mode
Table 7.2: Advantages and Disadvantages of X-as-a-Service Mode
Table 7.3: Advantages and Disadvantages of Facilitating Mode
Table 7.4: Team Interaction Modes of the Fundamental Team Topologies

CASE STUDIES & INDUSTRY EXAMPLES

Chapter 1
Industry Example: OutSystems (Part 1)—Miguel Antunes, R&D Principal Software Engineer, OutSystems
Chapter 2
Industry Example: Adidas—Fernando Cornago, Senior Director Platform Engineering, and Markus Rautert, Vice President

Platform Engineering and Architecture, Adidas
Chapter 3
Industry Example: OutSystems (Part 2)—Miguel Antunes, R&D Principal Software Engineer, OutSystems
Industry Example: IKEA—Albert Bertilsson, Solution Team Lead, and Gustaf Nilsson Kotte, Web Developer, IKEA
Case Study: Team-Focused Office Space at CDL—Michael Lambert, Head of Development, and Andy Rubio, Development Team Leader, CDL
Case Study: Stream-Aligned Office Layout for Flow-Based Collaboration at Auto Trader—Dave Whyte, Operations Engineering Lead, and Andy Humphrey, Head of Customer Operations, Auto Trader
Chapter 4
Industry Example: Spotify—Henrik Kniberg, Agile/Lean Coach, and Anders Ivarsson, Organizational Coach, Spotify
Industry Example: Feature Teams Supported by Cross-Subsystem Functions at Ericsson—Wolfgang John, Research Leader, Ericsson
Industry Example: DevOps Team Topologies at a Healthcare Organization—Pulak Agrawal, DevOps Manager and Technology Architect, Accenture
Case Study: Evolution of Team Topologies at TransUnion (Part 1)—Ian Watson, Head of DevOps, TransUnion
Chapter 5
Case Study: Strictly Independent Service Teams at Amazon
Case Study: Engineering Enablement Team within a Large Legal Organization—Robin Weston, Engineering Leader, BCG Digital Ventures
Case Study: Sky Betting & Gaming—Platform Feature Teams (Part 1)—Michael Maibaum, Chief Architect, Sky Betting & Gaming
Case Study: Evolving Highly Responsive IT Operations at Auto Trader—Dave Whyte, Operations Engineering Lead, and Andy Humphrey, Head of Customer Operations, Auto Trader

Chapter 6
- Case Study: Finding Good Software Boundaries at Poppulo—Stephanie Sheehan, VP of Operations, and Damien Daly, Director of Engineering, Poppulo

Chapter 7
- Case Study: Team Interaction Diversity at IBM around 2014—Eric Minick, Program Director for Continuous Delivery, IBM

Chapter 8
- Case Study: Adoption of Kubernetes to Drive Organizational Change at uSwitch—Paul Ingles, Head of Engineering, uSwitch
- Case Study: Evolution of Team Topologies at TransUnion (Part 2)—Dave Hotchkiss, Platform Build Manager, TransUnion
- Case Study: Sky Betting and Gaming—Platform Feature Teams (Part 2)—Michael Maibaum, Chief Architect, Sky Betting & Gaming

FOREWORD

Keeping our systems small and simple is a worthy goal, yet it is also one that most successful systems defy. Lehman's laws of software evolution, and, in particular, continuing growth, captures the evolutionary pressure to add capabilities as systems are used and new demands or opportunities are perceived. Being able to cope with, and even harness, this increasing complexity raises the importance of dual design arenas: the design of systems and the design of the organization that creates and evolves systems. We have a considerable body of work focused on the former; that is, on systems and software design and architecture, including an ever growing number of books on domain driven design and software architecture. Team Topologies addresses the design of the software development organization, with Conway's law in view.

> The basic thesis [. . . .] is that organizations which design systems [. . . .] are constrained to produce designs which are copies of the communication structures of these organizations. We have seen that this fact has important implications for the management of system design. Primarily, we have found a criterion for the structuring of design organizations: a design effort should be organized according to the need for communication.[1]

The above quote from the conclusion of Mel Conway's classic paper on organizational design for software development is a most fitting beginning to this book. Team Topologies describes organizational patterns for team structure and modes of interaction, taking the force that the organization exerts on the system as a driving design concern.

As the complexity of the system increases, so, generally, do the cognitive demands on the organization building and evolving it. Managing

cognitive load through teams with clear responsibilities and boundaries is a distinguishing focus of team design in the Team Topologies approach. To achieve duly scoped, bounded responsibilities, natural—and relatively independent—system (sub)structure is sought to align teams to. This takes Conway's law into account and leverages it to help maintain cohesive structures with clear boundaries and loose coupling (known as the reverse Conway maneuver, and described herein).

If this was the extent of it, Team Topologies would be a useful elaboration of Conway's paper, setting it in the current context. Of course, Team Topologies is even more than that. Notably, it identifies four team patterns, describing their outcomes, form, and the forces they address and are shaped by. Stream-aligned teams are the primary team form. These are teams that are optimized for flow, with all they need to effect continuous delivery of value and be fully responsive to the associated feedback cycles. This means that system design seeks not just loose coupling but a decomposition that supports flow and lowers dependencies and coordination needs between stream-aligned teams. Complicated-subsystem and platform teams reduce load for stream-aligned teams, where the latter are internal customers of the former's subsystem or platform capabilities (supporting all phases of development, delivery, and operations for multiple stream teams). Enabling teams likewise serve other teams, but they are service providers, helping stream-aligned teams learn new techniques, investigate new technologies, and so forth, allowing stream-aligned teams to retain focus while growing effectiveness.

Matthew Skelton and Manuel Pais have brought their considerable experience to bear, describing what these various team forms need to be successful, but also highlighting variations in context, identifying the design implications thereof, and indicating anti-patterns to avoid. They also, with great generosity, weave in insights from and offer pointers to related work. This, along with a set of case studies, further textures the book.

Team Topologies informs and enriches our understanding of organizational architecture, via the nuanced presentation of these key structural patterns, interaction modes or dynamics, and considerations for

evolution. And, due to its clarity and focus, it serves as a pragmatic guide whether forming teams and enabling them to meet their challenges or helping existing teams become more effective at responsive value delivery.

—Ruth Malan, Architecture Consultant at Bredemeyer Consulting

PREFACE

> [Modern] organisational design . . . is about designing for collaborative technologies, for the voice of the customer.
>
> —Naomi Stanford, Guide to Organization Design

Teams are always works in progress, but they are also your best shot at delivering value continuously and sustainably by aligning them with the business. Ideally, teams should be long lived and autonomous, with engaged team members. However, teams don't live in isolation. They need to understand how and when to interact with each other. And these team interactions need to evolve over time to support the distinct phases of discovery and execution that products and technology go through during their lifetimes. In short, organizations not only need to strive for autonomous teams, they also need to continuously think about and evolve themselves in order to deliver value quickly to customers.

This book offers a practical, step-by-step, adaptive model for organizational design that we have used and seen work across businesses at varying levels of maturity: Team Topologies.

However, Team Topologies is not a universal formula for building and running software systems successfully. There are teams and organizations who succeed with organizational dynamics very different from those described and recommended here (particularly in organizations with excellent culture and best practices already in place).

Team Topologies is meant to provide clear patterns that are straightforward for many different teams and organizations to follow and interpret, not to dictate to outstanding players how to perform. We like to think of Team Topologies as a set of music parts for an orchestra or big

band, not the melody for a top jazz trumpeter. Printed music for a large musical ensemble helps the group to succeed but does not dictate every aspect of performance; lots of detail is left for the ensemble to interpret to suit the occasion, venue, or mix of players. Likewise, there is huge value in agreeing to a coherent vocabulary and way of working together across teams to achieve good software delivery.

The Team Topologies approach helps organizations that are struggling to find a way to optimize their team structure, or for those that are not yet aware of the impact team design can have on good business outcomes and software systems in particular. Team Topologies helps organizations succeed more quickly and more continuously than before.

This book is for anyone who cares about the effectiveness of the delivery and operations of software systems: C-level leaders (including CTOs/CIOs, CEOs, CFOs, and so on) managers, heads of department, software architects and systems architects, and anyone else involved in building or running software systems who wants or needs to make the delivery and running of those systems more effective.

How We Came to Write This Book

In 2013, while introducing DevOps and Continuous Delivery at a company in the UK, Matthew devised the original DevOps Topologies patterns (and anti-patterns) in a blog post titled "What Team Structure Is Right for DevOps to Flourish?"[1] At the time, the company he was consulting with was struggling to adopt modern approaches to software delivery, and the early topology patterns Matthew created provided the company a way to explore different options.

Manuel interviewed Matthew at the QCon London software development conference back in 2015, where Matthew was speaking on Conway's law and the early DevOps Topology patterns. The resulting article, "How Different Team Topologies Influence DevOps Culture," was published by InfoQ and translated into several languages.[2] Later that year, Manuel helped to expand the DevOps Topology patterns and there were

contributions from the community.

Since then, the use of DevOps Topology patterns has exploded. They have been referenced over and over again in talks, articles, and conversations. They have helped organizations of all sizes and from varying industries around the world to think about the relationships between teams and how their interactions influence both organizational culture and software architecture.

Over time, we realized that the original DevOps Topologies presented a static view of team interrelationships that, while useful for initial discussions, was quite limited in scope. Through our combined experience with training and consulting organizations from across the world, we discovered that some teams work better relatively isolated or autonomous, while other teams work better with strong collaboration. We asked ourselves why, and we kept evolving our ideas based on feedback from our clients.

Eventually, this led to the Team Topologies as you see them presented in this book: a dynamic and evolving approach to organizational design based on real scenarios from across different geographies and industries.

How to Use This Book

Team Topologies is meant to be a functional book. It is our intention to provide content that is interactive and delivers as much learning as we are able to fit within these pages. To help with that, we have made some design choices that will help you navigate this book.

First, the book is organized in three parts:

Part I of the book explores Conway's law, the way organizational interrelationships constrain the design of systems we build, and how we can use this tendency to our advantage. We then define what we mean by teams and look at some practical constraints that affect effective teamwork.

In Part II, we investigate a set of static team patterns that have been proven in the industry and the implications of choosing one pattern over