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

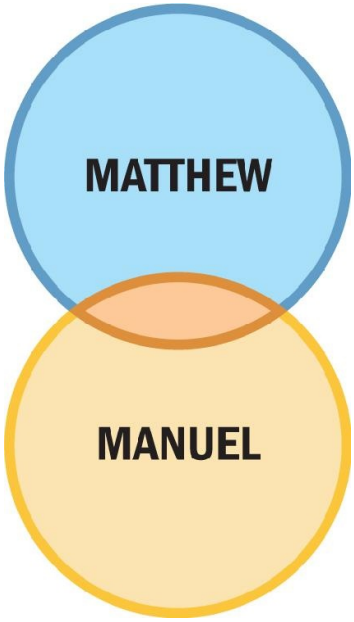
Cover and book design by Devon Smith

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

PART III EVOLVING TEAM INTERACTIONS FOR INNOVATION AND RAPID DELIVERY

FIGURES & TABLES

FIGURES

TABLES

CASE STUDIES & INDUSTRY EXAMPLES

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 another with Conway's law and organizational context in mind. This section should help you think about team topologies that are broadly suitable for your organizational

context. This part also provides some guidance in deciding how to align teams to areas of the system, taking into account Conway's law and fundamental team topologies.

Finally, in Part III, we deal with ways to evolve the organization design to provide powerful capabilities for innovation and rapid delivery in response to a quickly changing operating context. We explain how to use the Team Topologies approach to create a sensing organization that responds to the market and user demands, and accounts for the implications this has for hiring and skills.

Each part opens with a breakdown of key takeaways from each of the chapters. Throughout the chapters, we have included figures and callouts to highlight information we think is helpful to know and/or reference. We also provide easy-to-recognize scenarios, case studies, and explicit recommendations for different situations along the way.

Finally, the shapes, colors, and patterns found within many of the figures also have consistent meaning throughout much of the book. Here is the key:

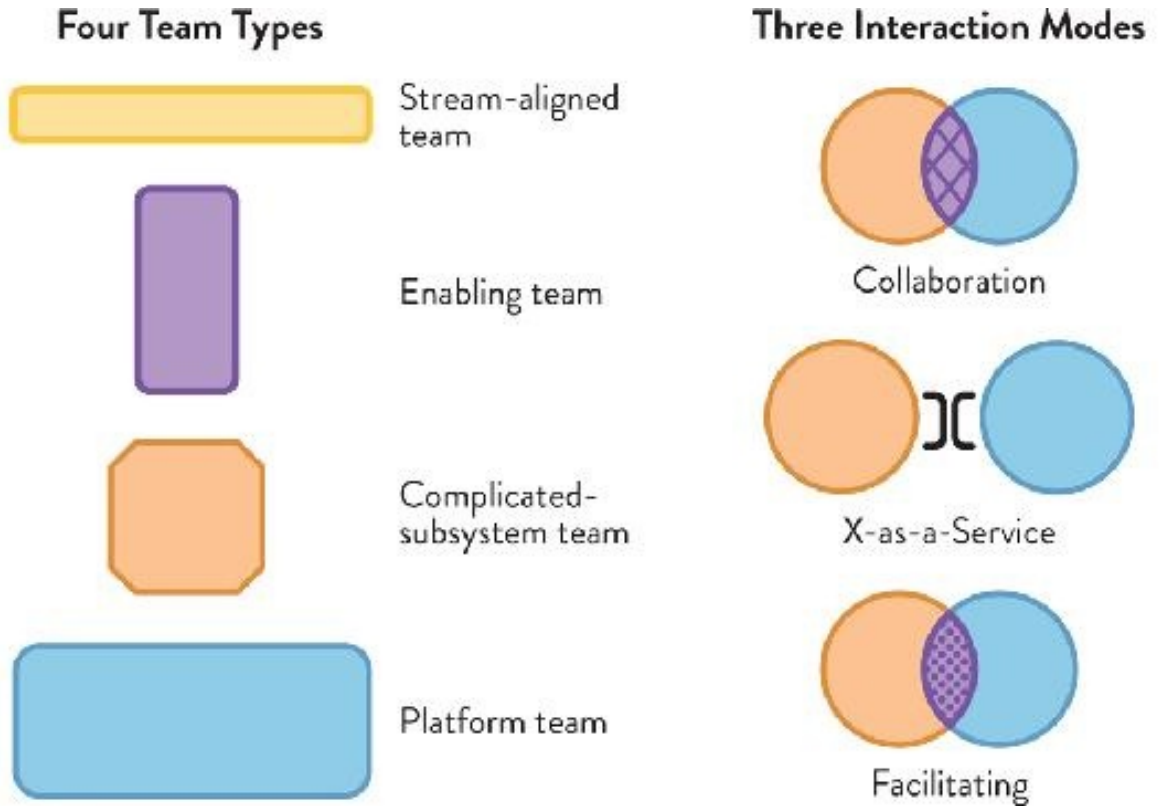


Figure 0.1: The Four Team Types and Three Interaction Modes

For the fullest understanding, you should read the book from cover to cover, as the subject matter builds up chapter by chapter. However, we have written the material so that each section is fairly independent.

In that spirit, here are some scenarios with corresponding ways to read the book that might match with your current situation:

- I need more clarity about different team types and which team types are effective.
  - Review Chapter 1 (overview), then Chapter 4 (static topologies), then Chapter 5 (fundamental topologies).
- I need to split up a large, monolithic software system.
  - Review Chapter 6 (boundaries) and then Chapter 3 (the team).
- I need to improve the architecture of the software system.
  - Review Chapter 2 (Conway's law), then Chapter 4 (static topologies), then Chapter 6 (boundaries).
- I need to improve the effectiveness of software development teams.
  - Review Chapter 3 (the team), then Chapter 6 (boundaries), then Chapter 5 (fundamental topologies).
- I need to improve morale and effectiveness within teams.
  - Review Chapter 3 (the team) and then Chapter 5 (fundamental topologies).
- I need to understand where to invest effort to help with projected growth.
  - Review Chapter 1 (overview), then Chapter 5 (fundamental topologies), then Chapter 8 (topology evolution).
- I need to understand how to evolve team topologies to meet changing business needs.
  - Review Chapter 7 (dynamic aspects) and then Chapter 8 (topology evolution and organizational sensing).

Kev Influences that Informed this Book

In addition to our own experience, this book is strongly influenced by several related approaches and sets of thinking. First, we assume that an organization is a sociotechnical system or ecosystem that is shaped by the interaction of individuals and the teams within it; in other words, that an organization is the interaction between people and technology. In this aspect, the book fits with ideas from the fields of: cybernetics (especially the use of the organization as a "sensing mechanism," which goes back as far as 1948, when Norbert Wiener's book Cybernetics: Or Control and Communication in the Animal and the Machine was first published), systems thinking (particularly the work of W. Edwards Deming), and approaches such as the Cynefin framework for assessing domain complexity (described by Dave Snowden and Mary Boone in their 2007 Harvard Business Review paper titled "A Leader's Framework for Decision Making"), and adaptive structuration theory (a term coined by Gerardine DeSanctis and Marshall Scott Poole in their Organization Science article, "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," where they emphasized that the impact of technology is not a given, as it depends on how groups and organizations perceive it).

Second, we assume that "the team" is something that behaves differently from a mere collection of individuals, and that the team should be nurtured and supported in its evolution and operation. In this respect, we draw on ideas from Bruce Tuckman (who proposed the four-stages model—forming, storming, norming, performing—for team development in his 1965 paper "Developmental Sequence in Small Groups"), Russ Forrester and Allan Drexler (who explored team-based organization performance in their 1999 paper "A Model for Team-Based Organization Performance"), Pamela Knight (who found evidence that storming takes place throughout the entire lifetime of a team in her 2007 paper "Acquisition Community Team Dynamics: The Tuckman Model vs. the DAU Model"), Patrick Lencioni (who explores common interaction issues in his seminal book The Five Dysfunctions of a Team: A Leadership Fable), and similar team-focused theories and research.

Third, we assume that Conway's law (or a variant of it) is a strong driver of software product shape and that organizations would benefit from explicitly

addressing the implications of this law. In this regard, we draw on writing and ideas from Mel Conway; from software architecture consultant and team organization design award-winner Ruth Malan; from ThoughtWorks technical director and one of the “reverse Conway maneuver” proponents James Lewis; and from similar authors and practitioners.

Finally, we draw on numerous sources that describe practical successes developing and running software systems at scale, including organizations such as Adidas, Auto Trader, Ericsson, Netflix, Spotify, TransUnion, and others. The size and speed of these organizations has made it possible for them to see tangible gains from changes in organization structure and team interaction over the space of several months to a few years.

As you travel through this book, we hope you get inspired to challenge how you think about teams, their structures, and how they function.

PART I
Teams As the Means of Delivery

KEY TAKEAWAYS

CHAPTER 1

- Conway’s law suggests major gains from designing software architectures and team interactions together, since they are similar forces.
- Team Topologies clarifies team purpose and responsibilities, increasing the effectiveness of their interrelationships.
- Team Topologies takes a humanistic approach to building software systems while setting up organizations for strategic adaptability.

CHAPTER 2

- Organizations are constrained to produce designs that reflect communication paths.
- The design of the organization constrains the “solution search space,” limiting possible software designs.
- Requiring everyone to communicate with everyone else is a recipe for a mess.
- Choose software architectures that encourage team-scoped flow.
- Limiting communication paths to well-defined team interactions produces modular, decoupled systems.

CHAPTER 3

- The team is the most effective means of software delivery, not individuals.
- Limit the size of multi-team groupings within the organization based on Dunbar’s number.
- Restrict team responsibilities to match the maximum team cognitive load.
- Establish clear boundaries of responsibility for teams.
- Change the team working environment to help teams succeed.

1 The Problem with Org Charts

> Organizations should be viewed as complex and adaptive organisms rather than mechanistic and linear systems.
>
> —Naomi Stanford, Guide to Organisation Design

Technology workers are in a constant state of action: creating and updating systems at an unbelievable rate, and combining different types of technology to create a compelling user experience. Mobile applications; cloud-based services; web applications; and embedded, wearable, or industrial IoT devices all need to interoperate effectively to achieve the desired business outcomes.

Today, these systems affect nearly every aspect of people's day-to-day lives in ways that are increasingly profound. If software is poorly designed—or more importantly, if there is a mismatch in the interaction of the software, the provider, and the customer—people will be adversely affected. They can be stranded long distances from home if a taxi-hailing application fails. They may be unable to pay rent if the software or processes for internet banking fail. They may even see their life in danger if a medical device fails. Never before has explicit sociotechnical design been so important.

Building and running these highly complex, interconnected software systems is a team activity, requiring the combined efforts of people with different skills across different platforms. In addition, modern IT organizations must deliver and operate software systems rapidly and safely, while simultaneously growing and adapting to changes and pressures in the business or regulatory environment. Businesses can no longer choose between optimizing for stability and optimizing for speed.