

**REMOTE TEAM
INTERACTIONS
WORKBOOK**

USING TEAM TOPOLOGIES PATTERNS
FOR REMOTE WORKING

REMOTE TEAM INTERACTIONS WORKBOOK

MATTHEW SKELTON AND MANUEL PAIS

IT Revolution
Portland, Oregon



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

Copyright © 2022 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.

First Edition

Printed in the United States of America

27 26 25 24 23 22 1 2 3 4 5 6 7 8 9 10

Cover and book design by Devon Smith

ISBN: 9781950508617
eBook ISBN: 9781950508624
Web PDF ISBN: 9781950508631

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

REMOTE TEAM INTERACTIONS WORKBOOK

CONTENTS

Preface	vii
Introduction	ix
A Brief Overview of Team Topologies	ix
How to Use This Workbook	x
Chapter 1 Overview: Focus on Remote Team Interactions	1
What Does an Organization Need in Order to Thrive in a Remote-First World?	1
<i>Resource: Cognitive Load Assessment</i>	3
Use the Team API Approach to Define and Communicate Responsibilities and Team Focus	4
Track Dependencies Using Simple Tools and Remove Blocking Dependencies	4
Overcommunicate Using Just Enough Written Documentation	5
Summary: Design and Define the Ways in Which Teams Interact	6
Chapter 2 Team Dependencies	7
Team API	7
<i>Resource: Team API Exercise</i>	11
Tracking Dependencies	14
<i>Resource: Team Dependencies Tracking Exercise</i>	17
Building Networks: Coffee, Talks, Internal Conferences	18
Chapter 3 Setting Team Boundaries	23
Group Trust Boundaries	23
<i>Resource: Trust Boundaries Exercise</i>	26
Online Space Setup	27

<i>Resource: Online Space Assessment</i>	32
Team-Focused Conventions for Chat Tools	34
Chapter 4 Purposeful Interactions	37
Team Interaction Modes: A Review	37
Listening to Team Interactions	43
Clarity Communication Purpose and Channels	47
Ensuring Clarity of Purpose of Platforms and Services	49
<i>Resource: Example Wiki Page: Thinnest Viable Platform Template</i>	53
Chapter 5 Next Steps	55
Design and Conduct a Developer Experience Platform Survey	55
Define Naming and Usage Conventions for Chat Tools	57
Use the Team API with Multiple Teams to Define and Clarify Team Boundaries	58
Devise and Share an Execution Plan	58
List of Resources	61
Notes	63
About the Authors	67
About <i>Team Topologies</i>	68

PREFACE

Why We Wrote This Workbook

The COVID-19 pandemic of 2020 and beyond has ushered in a new remote-first world for IT along with most other departments in the business. But many organizations have struggled to catch up with new tooling and ways of working. While some companies have embraced this new reality—ditching their expensive downtown offices and telling staff they can work from home permanently—many more organizations are discovering for the first time that the physical office was covering up poorly defined teams and poorly defined areas of focus, threatening their DevOps transformation efforts and the overall health and success of their business.

A successful remote-first approach requires the explicit design of communication between teams using physical and online spaces. Using simple tools for dependency tracking and patterns from *Team Topologies*, such as the team API, organizations will find that well-defined team interactions are key to effective IT delivery in the remote-first world.

In this workbook, we explore several aspects of team-first remote work, including:

- how the new “remote-first” world is highlighting existing poor team interactions within organizations
- why organizations should use the team API pattern to define and communicate the focus of teams
- how organizations can track and remove team-level dependencies
- how and why organizations should design inter-team communications consciously
- why and how organizations can use the three team interaction modes from *Team Topologies* (collaboration, X-as-a-Service, and facilitating) to help.

We hope these ideas and patterns will help you and your organization become more effective with a team-based, remote-first approach to building and running software systems.

Also, we'd like to thank Rich Allen for his invaluable contributions to this workbook.

—Matthew Skelton and Manuel Pais
August 2021

INTRODUCTION

A Brief Overview of Team Topologies

Team Topologies is the leading approach to organizing business and technology teams for fast flow, providing a practical, step-by-step, adaptive model for organizational design and team interaction. The Team Topologies ecosystem of partners, practitioners, and learning academy is transforming the approach to the digital operating model for organizations around the world.

In the Team Topologies model, four fundamental types of teams and three core team interaction modes combine with awareness of Conway's Law, team cognitive load, and responsive organization evolution to define a no-nonsense, team-friendly, humanistic approach to building and running software systems.

The four fundamental types of teams are:

- **Stream-aligned team:** aligned to a flow of work from (usually) a segment of the business domain.
- **Enabling team:** helps a stream-aligned team to overcome obstacles. Also detects missing capabilities.
- **Complicated-subsystem team:** where significant mathematics, calculations, and technical expertise is needed.
- **Platform team:** a grouping of other team types that provide a compelling internal product to accelerate delivery by stream-aligned teams.

READ MORE

You can read more about the four fundamental types of teams in *Team Topologies* on pages 79-110.

There are three ways in which teams should interact (interaction modes):

- **Collaboration:** working together for a defined period of time to discover new things (APIs, practices, technologies, etc.).
- **X-as-a-Service:** one team provides and one team consumes something “as a service.”
- **Facilitating:** one team helps and mentors another team.

READ MORE

You can read more about the three interaction modes in *Team Topologies* on pages 131–152.

The basic principles behind *Team Topologies* help organizations take a team-first approach to help unblock flow.

RESOURCE

You can also download the “Team Topologies in a Nutshell” and “Getting Started with Team Topologies” infographics at TeamTopologies.com.

How to Use this Workbook

We begin this workbook with an overview of the mindset and skills you and your organization will need to succeed in a remote-first world. The three main chapters (2, 3, and 4) each feature three patterns for improving team-based work in a remote-first context. Each improvement pattern has some explanatory context along with an example and suggestions for how to try it in your organization (labeled *Now Your Turn*). Each improvement pattern also refers to a section of the original *Team Topologies* book to provide a more detailed explanation, like this:

READ MORE

Read more about setting up team-first physical and online/virtual spaces in *Team Topologies*, pages 50–55.

While it's not necessary to read *Team Topologies* to take advantage of the patterns in this workbook, for the best results we recommend that you take time to contextualize the patterns of this workbook in combination with the ideas in the *Team Topologies* book.

Many of the patterns in this workbook also reference templates and other resources. Where possible, we've recreated the templates and resources for you in this workbook. We also provide links to these templates and resources online, so you can use them to get started. These online resources are free to use and open to contributions and suggestions for improvements. These tips will be featured like this:

RESOURCE

Use the Trust Boundaries template at [GitHub.com/TeamTopologies/Trust-Boundaries-Template](https://github.com/TeamTopologies/Trust-Boundaries-Template).

We've also worked to show the various ideas or techniques in the workbook that are related so that it is easy for you to navigate and make connections. These relationships are shown like this:

Chapter 2 in this workbook has more details on team APIs.

We hope these callouts help you navigate and get the most out of this workbook.

CHAPTER 1

Overview—Focus on Remote Team Interactions

A remote-first way of working requires a new mindset from organizations. This overview chapter explores some of the techniques that can help organizations adopt an effective remote-first approach.

What Does an Organization Need in Order to Thrive in a Remote-First World?

Many organizations have found, to their dismay, that rolling out a new chat tool for staff working remotely does not magically make the organization remote-first. A viable remote-first approach needs more than just chat and video tools.

Certainly, tools are needed and useful, but for a successful digital transformation—whether colocated or remote-first—the organization also needs good psychological safety and an effective set of “ground rules” and practices for teams to use for working together.

An example of this is Google’s five keys to successful teams. As they lay out, “Who is on a team matters less than how the team members interact, structure their work, and view their contributions.”¹

They describe these five key dynamics as:²

1. **Psychological safety:** Can teams take risks without feeling insecure?
2. **Dependability:** Can teams count on one another?
3. **Structure and clarity:** Are there clear goals, roles, and execution plans?
4. **Meaning of work:** Is the work personally important to the team?
5. **Impact of work:** Does the team believe the work matters?

Clear ground rules and practices define ways of working, set expectations, and provide easy-to-recognize patterns and modes of behavior that make it easy for people to work in well-defined ways. In particular, well-defined team interactions clarify the relationships between different groups in the organization and the purpose of different activities. This in turn helps to minimize the cognitive load on teams and provides more “head space” for focusing on the most important aspects of work within the organization.

Broadly speaking, cognitive load is the amount of mental effort being used on a task or set of tasks. For teams, you can think of cognitive load as the collective amount of mental effort being used by the team.

READ MORE

You can read more on cognitive load as it pertains to teams in *Team Topologies*, pages 11-12 and 39-47.

Cognitive Load Assessment

You can use this survey template as a starting point to assess the overall cognitive load of your team.

Answer each question on a scale of 1 (very poor) to 5 (very good).

1. How is the experience of building your services? Things to consider: Is building a clear and repeatable task? Is it fast “enough”? What happens when builds fail? Are failures easy to diagnose?
2. How is the experience of testing your services? Things to consider: Is testing a clear and repeatable task? Is it fast “enough”? What happens when tests fail? Are failures easy to diagnose? Are test environments adequate? Are test environments easy to access/spin up/clean up/inject test data into?
3. How is the experience of deploying your services? Things to consider: Are deployments a clear and repeatable task? Do you know what the deployment strategy is? What happens when deployments fail? Is it possible and straightforward to roll back a failed deployment? Do you have access to the necessary logs to understand why a deployment failed and/or its current status?

4. How is the experience of operating your services? Things to consider: Do you know how each service is being monitored? Do you have access to the data? Are adequate alerts (few false positives) being sent? Are logs and information accessible and easy to find? Are data flows across services relatively easy to follow?
5. How is the experience of being on call for your services? Things to consider: Do you know what the incident response procedure is? Do you feel you have enough experience (either real or simulated) to deal with incidents without high levels of stress? Do you know who to reach out to for help during an incident when you're on call? Would you be anxious about a 3 a.m. outage? What about an incident in a service that hasn't been modified for months or years?
6. How is the experience of dealing with health industry regulations and compliance? Things to consider: Do you feel you have sufficient awareness to raise questions on changes that might require compliance oversight or at least a quick debrief? Are you confident that you know which industry regulations are of concern for your services? If yes, do you feel that this knowledge is being refreshed often enough?
7. Would you like to comment on your overall engineering experience?

Notice that questions #1 through #5 focus on the experience of building, testing, deploying, and supporting software services, so they are broadly applicable. However, question #6 is specific to an organization working in the healthcare industry. It is included as an example of the kind of context-specific questions you will need to use in order to assess other aspects that might be causing high cognitive load for your teams.

The point is that this form is just a starting point. You will need to adapt and expand it to your organization's specific needs.

RESOURCE

You can access an online form of this assessment here:
[GitHub.com/TeamTopologies/Team-Cognitive-Load-Assessment](https://github.com/TeamTopologies/Team-Cognitive-Load-Assessment).

Use the Team API Approach to Define and Communicate Responsibilities and Team Focus

So what approaches can organizations take to improve interactions between teams? In *Team Topologies*, we explain the concept of a team API. A regular API is an application programming interface, a technical term for the way one piece of software interacts with another piece of software programmatically. A given team's API is therefore a kind of specification for how other teams in the organization can and should interact with that team.

READ MORE

Read more on team APIs in *Team Topologies*, pages 47-49.

The team API covers a wide range of aspects, including:

- artifacts owned by the team (libraries, applications, services, etc.)
- versioning and testing approaches
- wikis and documentation
- practices and principles
- road map and priorities
- communication preferences (when/how)

By defining these aspects and making them easily discoverable by other teams, a team increases its clarity of purpose and helps other groups to understand how that team fits into the wider organization.

See Chapter 2 of this workbook for more on team APIs, including exercises/templates.

Track Dependencies Using Simple Tools and Remove Blocking Dependencies

All teams are part of a socio-technical system, and therefore depend on other teams at some point in time, to a greater or lesser extent. That means we should

be tracking dependencies between teams now and over time. Some dependencies might be fine today, but in a few months from now they will start slowing down the dependent team too much and we'll need to address it.

While ideally we might want to remove all dependencies, in practice we should identify which ones are problematic and should be removed, and which ones are “under control,” for now at least. A problematic dependency introduces significant delays, is too unpredictable, or increases work in progress (WIP) for the dependent team, slowing them down considerably.

A remote-first environment creates unique challenges in this space. It's impossible to simply walk up to the desk of someone on another team to ask about progress. But a constant stream of chat messages asking for status updates becomes a cognitive burden.

Instead of spending time waiting on other teams to finish their work, focus on tracking and then removing these in-flow dependencies. Books like *Making Work Visible* by Dominica DeGrandis explain useful techniques for visualizing team dependencies, many of which can easily be adapted to work in a fully remote context. DeGrandis also has a great article on *TechBeacon*, “How to Defrag Your DevOps Value Stream,” that can be useful.³

Chapter 2 of this workbook has more details and exercises on tracking and removing in-flow dependencies.

Overcommunicate Using Just Enough Written Documentation

In a remote work setting, it's vital to “overcommunicate.” As Meaghan Lewis outlines in her article on *TechBeacon*, communicating mainly through text messages or chats makes discerning tone and urgency challenging, not to mention problems arising from cultural differences in phrases. Also, navigating when and how to post (to an individual or the whole team) is new territory.⁴

Ultimately, in a remote-work world it is essential to be very clear all the time about *what* you are working on, *why* you're working on it, *how* your work is being completed, and *when* it should be completed by. Overcommunication feels almost like an externalization of your key decisions and reasoning so that people can easily reconstruct the sequence of thoughts that led you to your current work. It is vital to successful remote teams.

Overcommunication will take several forms: share small decisions in a chat tool, write up larger decisions or designs in a wiki or document, and even create a presentation or report to explain important concepts. Don't just rely on people seeing scrolling messages in the chat tool.

Because most human-to-human interaction in a remote-first organization or team will be via chat and text media (such as wikis, documents, and so on), it is essential to emphasize good written skills. It's not just about typing lots of text, though. The text needs to have context when seen by itself.

For example, chatting "Hi, what do you think?" requires a mental context-switch for the person reading the message. What does that question refer to? They might even have to scroll back through a series of chats to find the original ask. On the other hand, "Hi. So, do you think we should switch component A for component B due to the performance issues with A?" gives plenty of context for the reader. There's no need for them to hunt down the original question or chat thread.

Don't make it hard for people to discover meaning in written communications. Make the messages self-contained.

Summary: Design and Define the Ways in Which Teams Interact

Well-defined interactions are key to effective teams, and this is especially true for remote work situations. Team-focused conventions within chat tools and wiki documentation increase discoverability and reduce cognitive load on communications. For example, using a chat tool like Slack you can enforce predictable channel naming conventions. Plus, the chat tool's ability to search across channels increases discoverability.

By adopting clear ground rules and practices, like the team API and predictable chat tool naming conventions (discussed in Chapter 3), organizations can take advantage of remote-first ways of working to increase their chances of success.