Forewords by Martin Fowler *and* Darren Murph

The ASYNC-FIRST PLAYBOOK

Remote Collaboration Techniques for Agile Software Teams

SUMEET GAYATHRI MOGHE

Praise for *The Async-First Playbook*

"Async is the superpower the most successful distributed teams have already realized, so understanding the playbook for implementing async-first practices is a must-do for any leader moving into this type of operating system. Sumeet has done a phenomenal job outlining that here."

—Chase Warrington, Head of Remote, Doist

"Collaboration is going async. The advantages are just too great: global reach—which enables more inclusion—and time zone immunity. Sumeet excellently decomposes the issues. And by zeroing in on async leadership, he tackles the elephant in the async room: leadership. Yes, you can be an effective leader, asynchronously."

—Cliff Berg, Co-Founder and Managing Partner, Agile 2 Academy

"*The Async-First Playbook* is a must-read for anyone part of agile teams in today's fast-paced and distributed work environment. This insightful book shows how adopting an async-first approach can transform the way teams collaborate. The result—higher productivity, better inclusion, and most importantly, fun! The future of work is remote, and async-first collaboration is the key to unlocking its potential. Through concrete examples, this book provides a roadmap for embracing asynchronous communication, enabling teams to focus on deep work and stay in a state of flow. I recommend this book to anyone seeking to unlock the full potential of remote collaboration and achieve better outcomes for their team."

—Iwo Szapar, Co-founder & Head of Remote,
Remote-First Institute

"Asynchronous work, when implemented well, can help create a sustainable and calm way of working. It enables individuals to design their days around productivity and priorities, creating a happier and more human-centric workplace. *The Async-First Playbook* thoroughly examines effective asynchronous work practices' benefits (and the challenges, too). This playbook discusses fundamental prerequisites that should be in place before transitioning to an async-first approach, including leadership practices, potential challenges, and a starter kit to help you determine your next steps. If you seek to alleviate information overload and minimize meetings, *The Async-First Playbook* is a valuable resource."

—Lisette Sutherland, Director of Collaboration Superpowers
https://www.collaborationsuperpowers.com/

"At a time when organisations are experimenting with how to make the most out of remote work, Sumeet offers a step-by-step approach on how to make software development practices more inclusive and future-proof. Furthermore, the section aimed directly at managers and leaders also turns the book into a strong advocacy-tool for sustainable remote collaboration."

—Pilar Orti, Director of Virtual not Distant

"*The Async-First Playbook* is a comprehensive guide for teams and leaders looking to optimize their remote work practices for a distributed world. The pandemic helped us see that knowledge work really shouldn't have a place, and now it's time to recognize and embrace that this work can be better accomplished asynchronously, too. Location-independent work? That's the first step in the transformation. Have you tried time-independent work? Sumeet shows us how."

—Tyler Sellhorn, host, *The Remote Show* podcast

"*The Async-First Playbook* is not just another book on Agile practices. It's a game-changer for teams looking to optimize their collaboration through asynchronous communication. In this comprehensive guide, you'll find everything you need to know about the principles, frameworks, tools, workflows, and cultural changes necessary to embrace an async-first approach. Even though I've been working in a remote setup with an async-first approach for many years, I still found improvements that I'll implement with my team. I found especially valuable the chapter on "The Async-First Leadership Mindset," and I can't wait to share this book with my team."

—Oana Calugar, Collaboration Consultant, Mural

"Every software company has recently been forced to adapt to remote work, and subsequently workers have awakened to the possibilities afforded by more flexible working. Companies have been experimenting and iterating to find a new equilibrium between in-office, hybrid, and fully remote work, and there have been many fits and starts. What's been missing is a practical guide on how to make the most of this newly popular work style so that both employees and employers win. Sumeet's book perfectly fills the void, giving teams and leaders the tools they need to escape the trap of synchronicity and make the most of their newfound asynchronous flexibility."

—Patrick Sarnacke, Managing Director, Thoughtworks UK

"The COVID-19 pandemic has provoked many publications about remote and hybrid work models. While benefits such as improved work-life balance and diversity have been praised, concerns about weakened personal relationships and burnout are also significant. One crucial element is that, too often, the cons result from companies attempting to put in place novel models while retaining old office-centric practices. That is why I like this book. In *The Async-First Playbook*, Sumeet Gayathri Moghe offers a practical, hands-on approach to making remote work effective for software development businesses. He emphasizes defaulting to action and provides actionable recommendations for optimizing distributed work arrangements. As someone who has led an organization through a similar journey and had valuable conversations with Sumeet while doing that, I can attest that this book packages knowledge from real-world techniques with a successful track record."

—Matheus Tait, Managing Director, Thoughtworks Spain

"Building on commonly established working patterns that have been adopted by many in the agile software development movement, Sumeet deftly re-examines the goals of these practices, and identifies remote-friendly alternatives that can lead your team to new heights. Of particular note is the emphasis on honing both writing and reading skills, and on guarding the time to use them both effectively. Those looking to navigate this new frontier would be well advised to set some time aside and read this first!"

—Andy Yates, Head of Technology,
Corporate Strategy, Thoughtworks

"The pandemic forced the tech workforce all over the world to work remotely. And since then, the focus has been on overcoming the challenges of remote working. Sumeet argues that the opportunity is bigger—to work differently, adopting async-first approaches and practices, to increase productivity and creativity by creating space for deep work and better collaboration amongst teams. *The Async-First Playbook* is a complete practitioner's guide to learn and implement async ways of working for yourself, your team, and your organization. In a new distributed and remote working world, this book has the potential to unlock productivity, better teamwork, and allow for improved work-life balance. A total must-read for technology or knowledge workers."

—Sameer Soman, Managing Director, Thoughtworks India

"Want to boost productivity and cut costs in your engineering organization? Sumeet's new book offers practical, actionable recommendations for optimizing team performance and streamlining processes via asynchronous ways of working. In today's digital-first world, boards are taking notice of the skyrocketing cost of engineering. Sumeet's book provides a valuable perspective on this issue, offering strategies for achieving better results in less time and at a lower cost. Whether you're a seasoned engineering leader or just starting out, this is a must-read for anyone looking to drive change and improve their organization's performance in a highly remote setting."

—Sagar Paul, Head of Global Solutions, Thoughtworks

"*The Async-First Playbook* is the book that the distributed agile world has been eagerly awaiting for years! Sumeet's direct and to-the-point writing style hits hard and very convincingly challenges our traditional norms. The book stays true to the format of a playbook by providing recommendations that are highly actionable and will enable the stated impact. An amazing must-read for software professionals."

—Santosh Mahale, Director, Engineering Group | e4r
(Engineering for research)

"The COVID-19 pandemic has accelerated the adoption of remote work, and it is now here to stay. However, many companies are still struggling to adapt to this new reality. This is where *The Async-First Playbook* comes in. In this book, the author makes a compelling argument for why async-first practices are essential for modern knowledge work. He guides the reader through the foundational elements needed to create an async-first culture, from individual and team practices to leadership and management styles. The author also highlights the challenges that come with remote work and provides readers with practical advice on how to avoid common pitfalls. Whether you're a developer, manager, or senior leader, this book will provide you with the tools you need to succeed in the new world of work."

—Sunil Mundra, Org. Change and Transformation Leader,
CXO Advisor, Thoughtworks

THE ASYNC-FIRST PLAYBOOK

THE ASYNC-FIRST PLAYBOOK

REMOTE COLLABORATION TECHNIQUES FOR AGILE SOFTWARE TEAMS

Sumeet Gayathri Moghe

Addison-Wesley

Hoboken, New Jersey

Cover image: Mr Aesthetics/Shutterstock
Chapters 4, 9: Various emojis, Carboxylase/Shutterstock
Figure 4.2: Screen capture, Microsoft
Figures 6.1, 10.2, 23.1: Screen captures, Atlassian
Figures 9.1, 9.2, 29.3: Screen captures, Google
Figures 11.1-11.4, 12.1, 12.3: Screen captures, Slack Technologies, LLC
Table 15.1: Based on Gayathri Mohan, F*ull Stack Testing: A Practical Guide for Delivering High Quality Software*, O'Reilly Media, Inc., 2022
Figure 18.2: Screen capture, ADR-Manager
Figure 20.2: Screen capture, Readme.so
Chapter 22, page 203: Definition of *busywork* courtesy of Pearson Education, Longman Dictionary of Contemporary English, 6th Edition, (c) 2014. Reprinted by permission of Pearson Education Limited.
Figure 21.3: Screen captures, Thoughtworks, Inc.
Table 24.1: Based on Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, "The SPACE of Developer Productivity," *Queue*, Vol. 19, issue 1, ACM, 2021
Figures 29.1, 29.2, 29.4-29.7: Screen captures, Mural

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2023940413

ISBN-13: 978-0-13-818753-8
ISBN-10: 0-13-818753-3

$PrintCode

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content accurately reflects the histories and experiences of the learners we serve.
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview).

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

- Please contact us with concerns about any potential bias at https://www.pearson.com/report-bias.html.

This page intentionally left blank

To every worker who's had joy sucked out of their life by a pointless meeting.

This page intentionally left blank

CONTENTS

FOREWORD

Two decades ago in a workshop, we looked for a name for a new approach to software development. We settled on “agile,” but one of the suggestions was “conversational.” What I liked about “conversational” was that it emphasized how we were advocating a more collaborative approach to building software. Instead of tasks carved out by managers and architects, delivered by (usually poorly written) documents, we wanted people to talk to each other. Database engineers could collaborate with those coding business logic using that data. Programmers should talk to their users about what software would be most valuable for their work. Communication is the bloodstream of a software effort, and we wanted to recognize that and increase its flow.

A few years later, Thoughtworks, my employer, opened offices in India. We had embraced and pioneered this agile approach to software development, and it was showing great success. But how could this work with a development center in India? The close collaboration we were encouraging was one thing when it was just a walk down a hall to talk to someone; what did it mean when separated by thousands of miles and a dozen time zones?

We found ways to make things work, and in time, technology came to assist us further. By the early 2010s, we were making heavy use of video-call technologies for our meetings (a habit that served us well when COVID-19 hit in 2020).

Sumeet Gayathri Moghe has seen this activity first-hand, working with us in India through most of this time. He’s managed several teams working across space and time zones in highly collaborative ways. As he's done this work, he's been reflecting on how effective our techniques are and how to improve them.

The video call has helped meetings work better when people are sitting far from each other, but even when in the same room, meetings aren't always the best way to collaborate. Questions that come up are often not best answered off the cuff—people need time to reflect. I like writing not just to communicate, but as a tool to arrange my thinking into a coherent structure, even for something as simple as this Foreword. When working across time zones, asynchronous activities become even more valuable, as they reduce the necessity to work at inconvenient hours.

It's a common slogan to hear "meetings are bad—get rid of them." But that's a simplistic reaction to the frustration of meetings that are badly run or inappropriately used. Sumeet has reflected on what situations require meetings and when they can (and should) be replaced by asynchronous techniques. This book sets out what he's learned and is a guide to improving our collaboration even as we sit further apart.

—Martin Fowler

FOREWORD

We've come a long way. Pioneers like GitLab wrote the playbook for remote work over a decade ago, yet it took a global pandemic to open the world's eyes to the possibilities of remote work. Today, many organizations embrace what we call "location independence." This encapsulates the magic that happens when a business decouples results from physical geography. While this has transformed millions of lives and enterprises, it's only the start of an even greater revolution.

The future of remote work—or, dare I say, the future of work—is time independence. Since at least the advent of the internet, the majority of knowledge workers have remained bound to time. The way we design our lives and the dreams we allow ourselves to contemplate are constrained by a fixed reality: the rigidity of a workday.

What happens when we shake off that rigidity? What design principles can humans apply to their own lives when we leverage tools, software, AI, and workflows to achieve professional goals outside the strict bounds of time?

We begin to optimize for what matters most. We reprioritize our individual identity stacks—the layers that make us individually *us*. Businesses become stronger and more resilient to crises. They become magnets for the most talented people, those who value flexibility over all else and who generate otherworldly results as a measure of their gratitude.

The "async-first" mindset is indeed what powers this time-independent way of working.

It's the key to saying "yes" to more midday adventures with my toddler. It's the cornerstone of the nonlinear workday—a concept I detailed in my own book, *Living the Remote Dream*. At the same time, it's a way to drive high-performing teams—something we've espoused for years as part of the "TeamOps" philosophy at GitLab.

Change, however, is hard. Concrete, tangible advice is tough to come by. We need frameworks and recipes to ensure that the asynchronous collaboration movement takes hold and scales to the ends of the earth.

The Async-First Playbook is one such resource. This book is the bridge between principle and execution. As opposed to being everything to everyone, Sumeet has focused this book on teams that build software. This, I'm sure you'll notice, makes much of the advice in the book immediately actionable for you, the reader. The focus by no means alienates other readers, though. Technologist or not, everyone can learn something about working asynchronously from this book.

There's a lot at stake. If workplace leaders get this right, we'll create a more flexible, inclusive, connected planet. This is where this book shines. Sumeet doesn't just focus on the everyday tasks that such teams perform, but he also sheds light on the responsibilities of effective managers and leaders in an async-first environment. His part-guerrilla, part-advocate approach is a great way for the intrepid yet enlightened leader to drive change in their realm of influence while they gather evidence and arguments for broader organizational change. And all the advice in the book comes to life with tool suggestions, examples, and resources for your team.

To be successful at asynchronous ways of working is to discover more of what makes work great—flexibility, space for deep work, frictionless knowledge-sharing, a bias for action, diversity, inclusion, and the ability to scale. By removing constraints through the power of technology, we empower society to put its focus on areas like community, service, and progress.

The plays in this playbook will empower you to lead and build differently. All that you need is an open mind and a belief that the status quo isn't worthy of defining the future.

—Darren Murph, Future of Work Architect and VP,
Workplace Design & Remote Experience at Andela

PREFACE

Welcome!

In March 2020, the world of work was in a frenzy. The pandemic had pushed everyone back to the safety of their homes, and it felt like our jobs and work were at risk. In the knowledge working and consulting industry, we were fortunate to have a higher exposure to modern collaboration tools. So, we coped better than some other industries that weren't as sophisticated with tool usage. As the months rolled on, our comfort with remote work grew to where we are today—most knowledge workers want to work remotely most of the time.

My path to remote work was slightly different from that of my colleagues. In the years preceding the pandemic, I had played several roles where remote work was the norm. For example, with one of my clients, I was the only Indian consultant in a group of 50-odd technologists, most of whom worked across Central and Eastern Europe on a suite of different products. It made more sense for me to work from my home office in Pune, India, than to commute to my company office several kilometers away. Through sheer coincidence, I became proficient with remote work well before most of my company colleagues.

As you can imagine, my clients, colleagues, and I had different pathways to the remote software development experience. With trial and error and some dumb luck, I'd realized well before the pandemic that a meeting-centric approach to remote work is unsustainable in the long run. Of course, the ability to jump into conversations and to tap each other on the shoulder models the office environment that we're all used to, but the office was no paragon of productivity. Mimicking the "office on the cloud" with modern tools re-creates the problems of the office at a much higher rate than earlier. The casualties? Our sense of satisfaction with work and our productivity.

Yet, it wasn't until I switched to part of the company serving North American clients from India that I could see the full extent of this "office in the cloud" dysfunction. For those of you in other countries, I want you to think of winter in the northern hemisphere. It's a 13.5-hour time difference between India and the West Coast of the United States! There's no overlap during standard office hours between these time zones. Even the East Coast of the United States isn't much better. But to this day, many of my colleagues in the tech industry straddle opposing time zones to collaborate with each other. The casualty? Our work–life balance.

But we needn't normalize these casualties. It doesn't have to be crazy at work. The tech industry will continue to grow. And at its heart, tech needs access to talented people. And those people won't always be in the same time zone as us. So, I believe distributed working is an inevitable part of the future of work. In recognition of that inevitability, this book aims to help software development practitioners build high-performing, distributed teams. Along the way, I hope I can show you how to make your processes less stressful, more efficient, more fun, more inclusive, and more thoughtful.

WHO SHOULD READ THIS BOOK

I've written this book for practitioners and leaders with a basic understanding of agile software development. If you're already working on a distributed or remote team, you'll recognize many of the challenges I describe in the book. You'll also see how going "async-first" can address many of those challenges. You may also be a manager or a leader who wants to guide their team or organization to a remote-native way of building software. In this case, this book will provide you with a blueprint to organize your teams. If successful, your experiments may catalyze broader change for your companies and clients.

People who've read early drafts of this book have told me it could be useful for a wider audience as well. Many chapters of the book that deal with collaboration and management practices are domain-agnostic. Indeed, if you're a curious knowledge worker, you needn't be a technologist to get value out of the book. I leave it to you to find your adventure with the table of contents!

HOW I'VE STRUCTURED THIS BOOK

As the name suggests, I've written this book as a practical guide to remote-native software development. Asynchronous collaboration is a means to that end. However, if you care about being agile and not just doing agile, you may not follow every practice I describe in this book. So, it's perfectly fine to read only the chapters that apply to your work. This isn't a book that I expect you to read from cover to cover.

To present the plays of this playbook, I've organized them into six different parts.

- Part I, "Adapting to the New Normal," introduces the value of asynchronous collaboration in our day and age and teaches you how to prepare your team to shift to an async-first way of working.
- Part II, "Prepare to Go Async-First," addresses the basics you need in place through tools, skills, and protocols before you tinker with your work practices.
- Part III, "The Practitioner's Guide," is the chunkiest section of this playbook. It aligns techniques for remote software development, with chapters that map to common software development practices. You can read every chapter if you're curious. But first, read the chapters that relate to practices you already follow.
- Part IV, "Async-First Leadership," teaches you how to be a supportive leader for your async-first team. The techniques in this part of the book will apply to people who are already in management or leadership roles or those who aspire to play such roles.
- Part V, "Navigate the Pitfalls," recognizes a few common traps when working async-first. No team is immune to these, so the mitigating techniques I describe in these chapters should interest every reader.
- Part VI, "Bring It All Together," closes out the book with a set of tools that'll help you guide your async-first shift.

So, dip in and dip out of the book as you shift yourself to an async-first way of working. Keep it close at hand so that when you're ready to introduce a practice to your team, you can refer to the plays relevant to it.

THE COMPANION SITE

Writing this book began by sharing some of my ideas with a broad audience. The companion website for this book, https://www.asyncagile.org, will continue to be the place where I'll share and discuss future musings related to this topic. As I learn, I want to learn aloud with you.

In many of the chapters, you'll see references to resources, examples, and templates that I've created for you. You'll find them all at https://www.asyncagile.org/book-resources. I encourage you to bookmark this page, so you always have the link handy.

With all that said, welcome, thank you for joining me, and let's get started!

Register your copy of *The Async-First Playbook* on the InformIT site for convenient access to updates and/or corrections as they become available. To start the registration process, go to informit.com/asyncagile and log in or create an account. The product ISBN (9780138187538) will already be populated. Click Submit. Look on the Registered Products tab for an Access Bonus Content link next to this product, and follow that link to access any available bonus materials. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

ACKNOWLEDGMENTS

As a first-time author, I have many people to thank for bringing this book to life. First, I must thank the OGs, particularly pioneers such as Jason Fried and David Heinemeier Hansson and companies such as GitLab, which have been passionate advocates of remote and async work. I stand on the shoulders of such intellectual giants. Of them, there are many.

Closer to home, I consider myself fortunate to have worked at Thoughtworks, a pioneer of agile software development, particularly in IT consulting. To be perfectly honest, most of the ideas in this book stem from my experiences in working at Thoughtworks. I've curated several techniques in this book that I learned only because other colleagues at the company took the pains to teach me.

I must reserve a special word of thanks to colleagues like Santosh Mahale, Sagar Paul, Sameer Soman, and Thoughtworks India's Engineering Group, who afforded me the time to work on this book. Their support represents why Thoughtworks is such a welcome environment for sharing what we learn.

I wouldn't have started on the journey of creating this book had it not been for the guidance and advice I received from Martin Fowler. Not only did he help me validate the ideas for this book, but he also took time out of his busy schedule to read my rather messy manuscript. I'll always value the feedback he shared with me. The book is richer for it. Martin has also written a foreword for this book, which I consider a massive endorsement. Speaking of forewords, this book has two. The other one is from Darren Murph, one of the most popular remote-work leaders out there. Even while settling into a new role, Darren took the time to write a thoughtful note about the book. I can't thank him enough for it.

I also received tons of advice and encouragement from Gayathri Mohan, Paulo Caroli, Sunil Mundra, Vinod Sankaranarayanan, and Mangalam Nandakumar. As published authors, they were generous enough to share their experiences with me, so I had a frame of reference for what writing a book is like.

Sachin Dharmapurikar, Vijay Raghavan Aravamudhan, Sunita Venkatachalam, and Allison MacQueen gave me feedback on several pieces of writing. They've enriched the ideas in this book with their input. And how can I forget Dr. James Stanier? He was the technical reviewer for this book, and his candid, point-by-point feedback has helped me sharpen the book into the product you see today.

When ideas collide, new ideas emerge. Amy Luckey, Daniel Pupius, Mohammed Najiullah, and Priya Darshini M G helped me out by writing about their ideas and experiences of working and leading remotely. Their thoughts have inspired a lot of the content you see in the book.

Greg Doench, my editor at Pearson, was one of the first individuals to believe in the concept of this book, demonstrating unwavering faith from the start. His contagious enthusiasm has been a driving force throughout this project. Menka Mehta from Pearson India has been my go-to person throughout the development of this book. If I had a question, she always had an answer. Dr. Chris Zahn, the developmental editor for this book, offered me insightful written feedback that deepened my understanding of style guides and the process of crafting a polished manuscript. My copy editor, Kim Wimpsett, went through my manuscript with a fine-toothed comb. Her attention to detail helped me uncover many blind spots in my writing style. Julie Nahil and Charlotte Kughen guided me through production and design activities that I'd never experienced before writing this book. Without them and the entire Pearson brigade, this book wouldn't be in your hands.

As I take these ideas to more people, I must acknowledge the role of colleagues such as Pranav Shah, Ashwin Lal, Avneesh Chandhok, Ming Linsley, Shipra Shandilya, Jem Elias, and Jaydeep Chakrabarty for creating consulting and speaking opportunities for me. Even the idea of async-first work sometimes needs synchronous storytelling!

The last few years have taught me how valuable friends and family are. I must acknowledge those people in my life. Nagarjun Kandukuru has been a friend for the last decade and a half at Thoughtworks and an advocate for asynchronous

collaboration. His support for my work has been heartwarming. My close friends, Anuroop Krishnan, Manish Vaidya, Chetna Soni, and Surali Vaidya, have been among my biggest cheerleaders. And my lovely wife, Gayathri, has always been my first sounding board. She must be tired of listening to my ideas and reading every first draft of what I write, and yet she humors me. If you enjoy reading the book, it's because it first passed through her filters. My kids—Avni and Vihaan—don't know yet that this book was a labor of love for their dad. But they were understanding enough to leave me alone when I was writing. They'll know how much that has meant to me when they grow up. For now, I put my gratitude in writing. I promise to make up for any time I lost with them.

Finally, a big shout-out to everyone who followed and supported Asyncagile.org in its infancy. You know who you are. You needn't have patronized me at the time. And yet, you took the time to read what I had to say, and you spread the word around. Without you, I'd have never had the confidence to write this book. Thank you, whoever you are, and wherever you are.

This page intentionally left blank

ABOUT THE AUTHOR

Sumeet Gayathri Moghe is an agile enthusiast, product manager, and design nerd at Thoughtworks. He's worked with a variety of clients over the last few decades, building software products and helping them improve their engineering effectiveness. During his time as a consultant, Sumeet has exposed himself to various domains such as retail, travel, telecom, payments, healthcare technology, education, and more. That, in turn, has helped him generalize his experience across industries. What you see in this book stands for what he's learned from his colleagues, their successes, and their occasional misadventures.

Sumeet now lives in Pune, India, with his wife and two kids. When he's not at work building software, you'll probably see him looking through a camera's eyepiece, photographing wildlife or a scene in the wilderness.

To get in touch, find him on Linkedin (@sumeetmoghe) or visit asyncagile.org where he blogs almost every week. If you're curious about his photography work, check out www.sumeetmoghe.com.

This page intentionally left blank

Part I

ADAPTING TO THE NEW NORMAL

The COVID-19 pandemic forced the knowledge work industry into remote work. Today several companies and teams continue to work remotely, though they still follow office-centric work practices. In this part of the book, I argue that these ways of working must change so we can take advantage of our new, distributed work arrangements.

- **Chapter 1**, “There’s Got to Be a Better Way to Work,” sets the scene for how asynchronous work has the potential to transform distributed software development so it’s fun, sustainable, inclusive, and scalable.
- **Chapter 2**, “Foster a Mindset for Change,” teaches you how to prime your team for a shift to an async-first way of working. Before you explore the playbook, the ideas from this chapter will help you create a framework to align your colleagues to the change.

This page intentionally left blank

1

THERE’S GOT TO BE A BETTER WAY TO WORK

6:25 am Nita pulls herself out of bed, having pushed snooze on her alarm five times already. Oh boy, she’s late! She shakes her son, Abin, awake. They barely have 30-odd minutes before they drive to school. Dang it—she has to drive. She’d much prefer to take the scooter, but it’s pouring outside. If only she could control the Indian monsoon.

Okay, 30 minutes. Freshen up, make some coffee, and get Abin’s lunchbox ready. Drat! She forgot to order a milk delivery. Thank heavens she has some milk powder. She hustles to get a couple of sandwiches ready as she fries some eggs so Abin can have breakfast.

7:45 am That wasn’t the start Nita was hoping for. Life as a single mom is hard. She barely got her son to school on time. She’d have liked to wake up early and stay on top of things, but she went to bed late and didn’t get a good night’s sleep. Work doesn’t begin till noon. Maybe she can sleep now? Or maybe not. She’s got to cook, get ready, take out the trash, and put away the dishes. That won’t happen by itself, will it?

12:00 pm Nita works a midday shift. Work at home in Bangalore begins at 12 pm and, with an hour’s break thrown in, ends at 10 pm. She works for a client in Boulder, Colorado, and some of her American teammates are based out of Boise, Idaho. This is the only way she gets an hour to overlap with them for meetings. She tried getting 30 minutes of shut-eye before starting work, but she ended up just tossing and turning in bed. The day has barely begun, and she already feels tired.

2:30 pm Time to pick up Abin. There goes 40 minutes of her break time. Lunch was already at breakneck speed—coffees and dinners will probably need to be at the worktable. Sigh.

5:00 pm As a designer, Nita is trying to build some wireframes for a new feature the team is scheduled to deliver in an upcoming sprint. But she has a meeting with the team's business analyst in 15 minutes. On some days, Nita wonders if she's getting any work done at all. The last four hours consisted of two hours of meetings—one meeting to onboard a new team member and another meeting to discuss the sprint planning meeting later this evening. She tried to make some progress independently, but there wasn't enough contiguous time or enough available information for her to start.

9:00 pm Abin got back from cricket coaching, showered, and warmed up dinner. He's a good kid. Before heading to bed, he hugged Nita and kissed her goodnight. "Tomorrow will be an on-time start," he promised. Nita's been powering her day with coffee. The team is now in a sprint planning meeting with the folks in Boise and the client in Boulder. One more hour to motor through. Thank heavens for that coffee.

10:00 pm Aargh! She's not done yet. The client asked for some changes to the proposed iteration plan. Nita now has to finalize wireframes for user stories that are kind of ready but not quite. The team huddle is in 15 minutes.

11:00 pm Finally, time to get some sleep. Uh-oh! Hold on. She's got to put the milk bag out and double-check the milk order. Abin wasn't as good a boy as she thought, either. He didn't clean the kitchen before heading to sleep. Time to roll up her sleeves and clean up.

She gets the job done, changes, brushes, flosses, and gets into bed. Try as she might, she just can't sleep. Too much caffeine in the system? Too much of that bright screen? Who knows? Her attention shifts to her phone. The team is discussing the change in sprint plans. Maybe she should tune in until she feels sleepy.

WORK DESERVES A NEW LOOK

Phew! That day, unreal as it may be, is commonplace for many remote-working technologists operating across continents. Nita's character is fictional, but I know plenty of people who live and work like her. Of course, work hours can differ depending on the time zones the team is distributed across. But context-switching and highly interrupted days are common regardless of time zones.

In 2020, the global pandemic sped up the shift to remote work. To be sure, that was a change in the right direction. In a time of powerful computers in our pockets, great tools for online collaboration, and reasonably good internet connectivity in most places, it was strange to make people go through hours of commuting just to be in a noisy office every day. People made considerable sacrifices in their personal lives to have a career. For those with a single-minded devotion to that career, it was fine—but there were others who were looking for balance in their lives. And so, remote work came with the promise of restoring that balance. It gave organizations a way to tap into talent pools they hadn't considered before, simply because they didn't have offices in that area.

If you can now work from anywhere, an organization can employ you from anywhere. Knowledge workers are always more in demand than the supply, so being able to widen the talent pool gives employers a way to scale despite the constraints.

The purpose of this book isn't to extol remote work, though. There are plenty of other books to do just that. The question I like to ask is about Nita—is there a better way to work? The shift to remote work in 2020 was so abrupt that many organizations never had the time to consider if the same work practices that felt effective in an office are also relevant for remote work. As a result, you see many individuals, such as Nita, continue to make compromises to their work–life balance, their mental and physical health, and their ability to do deep, meaningful work. There's got to be a better way to work.

This book articulates a new way of working, especially for software development teams. That way is to embrace *asynchronous collaboration*. "What is that?" you may ask. The best way to define it is to combine a few definitions I've derived from Catherine Tansey and Marcelo Lebre.*

* Citations of facts and sources appear at the end of the book. A page number and a phrase identify the passage.

> Asynchronous work is the practice of working on a team that doesn't require multiple people to be online simultaneously. You do as much as you can with what you have, write things up clearly, transfer ownership of the work to whomever needs to pick it up next, and then work on something else.

If that sounds quite different from the way you collaborate with your team today, let me explain why you should consider making the shift to this way of working.

COMPLEX PROBLEMS NEED SMARTER COLLABORATION

When I started my career in IT, the problems we solved for clients differed greatly from the problems we solve today. For example, few clients expect us to build simple create, read, update, delete (CRUD) applications—low code platforms have disrupted that space. How about simple storefronts? Services like Shopify have disrupted that space by simplifying it for noncoders. With services such as managing large data centers, the hyper scalers such as Azure, GCP, and AWS have made it easy for in-house IT teams to manage infrastructure.

The work we do today is far more complex—we build platforms and data meshes and deep learning and neural networks. We modernize decades-old legacy systems to give incumbents a way to compete with digital native disruptors. We explore new ways to develop the human–machine experience. It's impossible to run these kinds of projects with any of the following problems:

- An overdependence on meetings, conversations, and tribal knowledge
- A lack of good written communication
- No commitment to deep, uninterrupted work

Asynchronous collaboration comes with the promise to meet only when necessary, to make conversations productive, and to give people time so they can actually work.

THE PROMISE OF FLEXIBILITY

In April 2022, I surveyed more than 450 Indian employees in a leading software development unit of a global IT firm. I asked them what they dislike about their work. These were the top two responses from more than 50 percent of the respondents:

- Long contiguous hours
- Late hours

This is almost antithetical to the promise of remote work. Luke Thomas and Aisha Samake, in their book *The Anywhere Operating System*, outline this promise:

> When coworkers say, 'I love working from home,' it's about when they work instead of only thinking about where they work from. That's the secret hiding in plain sight!

Indeed, less than 5 percent of the people I surveyed recently want to go into an office every day. Sixty percent of the respondents preferred to work "regular, daylight hours," i.e., nine to five. Twenty-eight percent wanted a flexible schedule where they could do their eight hours, their way. And these expectations are changing. This observation bears itself out in Future Forum's research, which surveys a broader audience than my own:

> Knowledge workers who say they have little to no ability to set their own hours are 2.6x as likely to "definitely" look for a new job in the next year (compared to those with moderate schedule flexibility).

While the job market and economic conditions at any point may or may not afford such opportunities, it's clear where people's preferences are. Asynchronous work allows people the autonomy and flexibility they desire by allowing them to work during the hours that are most productive for them. More important, it lets managers and employers project a shared sense of trust and empowerment.

ESCAPING THE SHALLOWS

As a product manager and an aspiring designer, I was always at sea in an open office. If the constant white noise wasn't bad enough, the fact that I was "right there" implicitly meant that I was available to be interrupted. I got little joy out of my working day, especially when I needed to get work done for the team and needed time with my head down. I craved what Cal Newport terms *deep work*—the ability to concentrate without distraction on a cognitively demanding task.

When we started working remotely, though, I could cut out the white noise. Suddenly, I had my corner office. Except, of course, it was a room in my house. What didn't change were the interruptions. The across-the-table interruptions now became video meetings and all-day back-and-forth messages on our instant messaging platform. Like Nita, I've struggled to get time to do deep work because my day would get cut up into little one-hour chunks. Paul Graham noted this eloquently in his 2009 essay "Maker's Schedule, Manager's Schedule":

> When you're operating on the maker's schedule, meetings are a disaster. A single meeting can blow a whole afternoon, by breaking it into two pieces, each too small to do anything hard in.

Asynchronous collaboration advocates for "meetings as the last resort, not the first option." If your current collaboration practices need loads of meetings, I understand this sounds radical. Asynchronous collaboration advocates for using communication tools thoughtfully. The sentiment I'd like you to empathize with is that interruptions have a cost—not just in terms of the time spent in the interruption but also as a fallout impact on productivity. Mihaly Czikszentmihaly, in his bestselling book *Flow: The Psychology of Optimal Experience*, speaks of a state that most of us want to get to at work:

> … a state in which people are so involved in an activity that nothing else seems to matter; the experience is so enjoyable that people will continue to do it even at great cost, for the sheer sake of doing it.

Here's the killer, though: while 97 percent of the 450-odd people I surveyed claimed to care about "flow," only 12.5 percent—i.e., one in eight people—claimed to achieve it with regularity. Asynchronous work promises to give people a better chance at achieving flow.

The other benefit of deep work is the ability to have deep interactions with co-workers. In his landmark book *Thinking, Fast and Slow*, Daniel Kahneman outlines two systems of thinking. System 1 is an intuitive and fast way of thinking. You call on System 1 for the answer to your parents' names, for the result of 2+2, and for how you navigate your own home. Anything that needs slightly deep thought—double-digit multiplication, for example—needs slower thinking. Kahneman calls that kind of thinking System 2. System 1, being as instinctive as it is, can help us decide at speed but is also prone to biases and errors. An effective approach to work should combine both System 1 and System 2 thinking.

A culture that defaults to meetings as a way of working lives in the realm of System 1. There's barely enough time to slow down, go a few levels deep, and analyze things. You lionize responsiveness and presence, but deep work becomes the casualty. Asynchronous work prioritizes good analysis and thinking. When you do meet, people have given enough thought to a topic and are meeting for a specific time-boxed purpose that needs intense collaboration. You can now make the deep work you've done so far deeper.

A MORE INCLUSIVE WORKPLACE

The people I surveyed were in offshore development teams in India. Their unit served clients in North America. Many people were straddling time zones with up to a 13-hour time difference. You don't need me to tell you that this is hard. And if something is hard for most people, it's usually harder for historically under-represented groups such as women in IT. (Even after years of conversations about the topic, women's representation in IT was just 33 percent in 2022. In technical roles, that goes down to 25 percent.)

For example, a tight schedule with loads of meetings isn't easy for anyone, but it's especially hard for women, who bear a disproportionate burden of household and childcare responsibilities in most societies. Now let's imagine a workplace without arbitrary start and end times, where people work in schedules that make sense for them and when we don't have the pressure to cram all communication into those specific eight hours. Getting rid of those constraints can help more under-represented groups find a way into the workplace. We can't change society overnight. What we can change is the workplace—and that's the promise of being asynchronous.

While we're on the topic, consider the diverse personalities in the workplace. There are introverts who will rarely be the first to voice their opinions on a topic. There are people who are non-native English speakers who may be deep thinkers but lack the confidence to articulate themselves. Asynchronous communication allows them to have the space to share their thoughts and ideas at a pace convenient for them. Modern writing tools allow non-native English speakers to correct their spelling and grammar. Introverts now don't have to be the loudest voice in the room. If diversity is being invited to the party and inclusion is being asked to dance, asynchronous work helps that inclusion.

DEFAULTING TO ACTION

In software engineering, there's always a choice between waiting to be perfect and being wrong at speed. When everyone was in one office, sitting at one large table, people could ask for or offer help by just tapping someone on the shoulder and talking to them. Never mind the fact that the person who needed to help you might have lost their own flow. Still, you were one step closer to being perfect. Using the same approach in a remote setup creates many problems. It's hard to keep track of who is free, who is busy, and who you're interrupting when you're giving each other a shout. Remote.com speaks of the concept of "defaulting to action" in such a situation. Sure, interrupting someone can help get your work closer to perfection, but you can't overlook the impact it has on the person who is unblocking you. Instead, you choose to be wrong at speed. It's okay, if you have to backtrack at a later point, but it's better than waiting to be unblocked.

> *There are many times when work isn't ready for us to tackle, tasks aren't planned, decision makers aren't online, etc. In these times, successful teams execute, even if they later have to refactor and adapt, they don't waste time "waiting."*
>
> —Marcelo Lebre

Defaulting to action optimizes for speed and throughput, but it also encourages thoughtful communication. For example, a well-written requirement may preempt the need for a ceremonial kickoff, where a developer, product owner, and tester discuss its implementation and testing approach. A simple screen recording attached to such a requirement can allow a developer to continue development while giving the product owner and the tester a way to review the implementation, without interrupting what those two might be up to. Sure, the trio may meet up to iron out some sticking points, but the mantra of "meetings as the last resort, not the first option" works well here. If the developer does have to backtrack, it's also feedback for the user story. The product owner can incorporate this learning into the next set of user stories they write. When in doubt, teams that adopt asynchronous communication just execute.

IMPROVED KNOWLEDGE SHARING AND COMMUNICATION

Asynchronous collaboration isn't black magic. Getting time back to do deep work isn't a freebie. In return, people must be clear communicators—mostly through writing.

Many of us find it daunting to write. I admit we all must get better at this skill, but for now let's focus on the potential benefits. Daily writing brings with it many advantages, including the following ones:

- A regular record of project decisions that helps anyone understand your project's current state effectively.
- Easier knowledge sharing and project ownership transfer.
- Reducing fear of missing out (FOMO) because everyone can now know what's happening in any part of the project. There's no need to attend a meeting to learn what happened in the last meeting!

As projects and organizations scale, tribal knowledge and a system of "he said, she said" don't scale. Effective writing and thoughtful curation have a better chance at helping you scale, even as tenured people leave and new people join your team.

A BETTER WORK ENVIRONMENT FOR ALL OF US

Like most practices, asynchronous collaboration isn't a silver bullet. It won't solve all your problems. Regardless, it's safe to assume that every team that does any creative work will benefit from embracing asynchronous practices. Figure 1.1 summarizes the benefits we've discussed in this chapter.

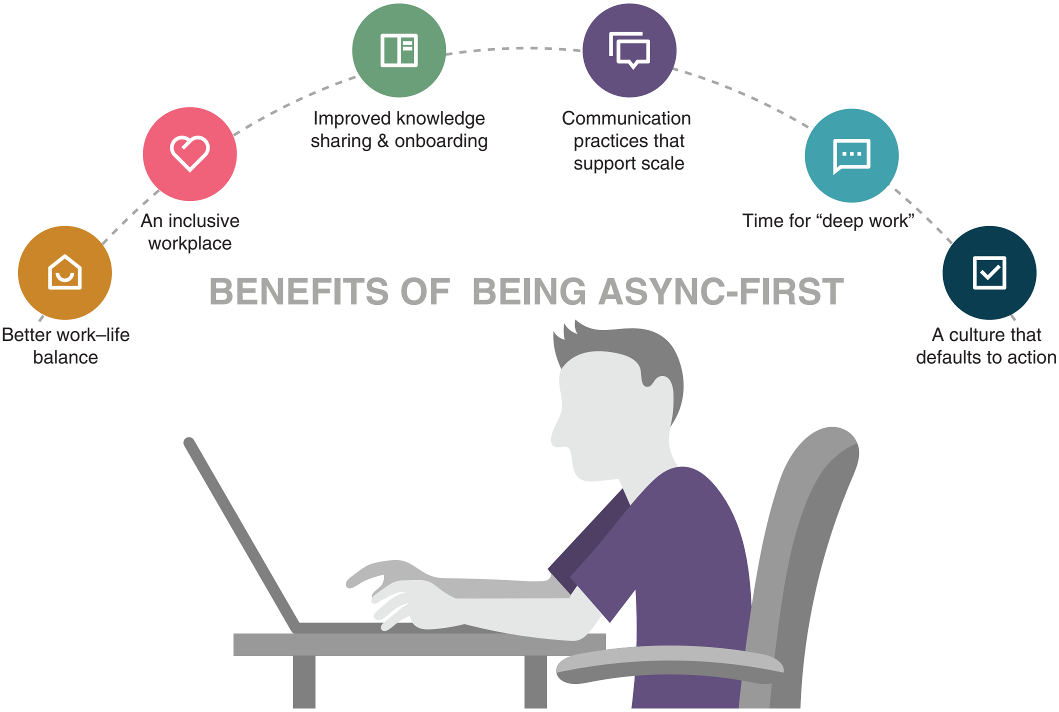
Improved knowledge sharing & onboarding
Communication practices that support scale
An inclusive workplace
Time for "deep work"
Better work–life balance
BENEFITS OF BEING ASYNC-FIRST
A culture that defaults to action

Figure 1.1 Benefits of an async-first work culture.

CHAPTER SUMMARY

The world of work has changed considerably since the start of the global pandemic, as many of our teams are now highly distributed and remote.

- Remote and distributed teams can experience burnout if they operate in primarily a synchronous manner.
- If you're building software or doing knowledge work in today's day and age, you limit your productivity if you depend too much on meetings and tribal knowledge. You need effective written communication and the ability to do deep work.
- The ability to work asynchronously, where people don't have to be in the same physical or virtual space simultaneously, has several benefits:
    - Better work–life balance
    - Higher inclusion
    - Improved knowledge sharing
    - Communication practices that support scale
    - Time for "deep work"
    - A culture that defaults to action

While the global pandemic of 2020 was a time of great pain for many of us, it also fast-tracked us into a new set of possibilities for how we work together in teams. How far you embrace these possibilities will depend on your appetite for change and your team's context.

Throughout the rest of this book, I'll help you identify asynchronous work techniques for your team. But before you try to change anything, you must enlist your team's support. So, in the next chapter, I'll describe how you can introduce this async-first shift to your immediate colleagues.

Index