Creating Agile Organizations

A Systemic Approach

Cesário Ramos | Ilia Pavlichenko

Forewords by Craig Larman and Dave West

FREE SAMPLE CHAPTER |

CREATING AGILE ORGANIZATIONS

This page intentionally left blank

CREATING AGILE ORGANIZATIONS

A SYSTEMIC APPROACH

Cesário Oliveira Ramos

Ilia Pavlichenko

Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Cover image: Sukpaiboonwat/Shutterstock

Figures 3.9, 3.11: Adapted from *This is Lean: Resolving the Efficiency Paradox* by Niklas Modig and Par Ahlstrom (2012).

Figures A.1, A.2, A.3, A.4: Adapted from *A Scrum Book* by Jeff Sutherland and James O. Coplien (2019).

Figure A.15: Adapted from *Thomas-Kilmann Conflict Mode Instrument* by Kenneth Thomas and R. Kilmann (2002).

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2022938926

ISBN-13: 978-0-13-585319-1
ISBN-10: 0-13-585319-2

ScoutAutomatedPrintCode

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 Jacqueline, whose patience and love continue to amaze me.
—Cesário Ramos

To my precious Olga and beloved children, Gordei and Evaliya, whose support I felt so much throughout the writing.
—Ilia Pavlichenko

This page intentionally left blank

Contents

This page intentionally left blank

Foreword

Perhaps the greatest misunderstanding about Agile—especially at scale—is that it's just a new *process or way of working* or *mindset* or *culture*. Senior management then incorrectly thinks, "Since this is just a change of practices and processes and mindset, our role is to *support* it with education for the hands-on people. And . . . these new practices and culture can be introduced into the *existing organization*."

From this ignorance flows a world of dysfunction, disruption, and degradation, leading to the widespread failure of Agile to provide high-impact and lasting benefits.

Cesário and Ilia are among a small group of coaches who really understand both this mistake and its remedy, focusing on a key element of successful change to being adaptive at scale.

Organizational Design

Cesário and Ilia also understand that organizational design (OD) is *not* something that senior managers should *support*. Rather, that OD is something senior managers need to themselves own, master, create, and lead—rather than something delegated to middle management or consultants. In that sense, this book is a practical primer on OD for senior management.

Also, near the top of the list in Agile misunderstandings is the *incorrect* belief that *Agile equals deliver fast*. Sure, early, and frequent delivery is important for learning, reducing risks, and making an impact (or not). But especially at scale, consider this scenario: There are two fast-delivery *feature teams*, each specialized in a different domain: one in bond trading and the other in equities trading. Suppose they both deliver fast, and it's discovered that there's little money (this quarter) in bond trading, but lots in equity trading. But—and this is the key point—as long as there exists a formal team called the Bond Trading feature team, and probably a Bond Trading software development manager, then even though they have learned "bonds currently suck," the team is likely to keep doing the same kind of work, because the *organizational design has formally locked them into working on that subject*, often under the local-optimization thinking mistake of "local efficiency."

So, fast delivery and learning aren't enough at scale. Without other adaptive OD elements in place—such as teams not formally dedicated to one specialist domain—there might not be any concrete change.

Cesário and Ilia understand that. And they help us understand all that in this wonderful book from two passionate people with years in the trenches involved in large-scale adaptive development.

—Craig Larman

This page intentionally left blank

Foreword

If surviving in the 21st century had a theme, it would be agility. As an organization, as a team, and as a human being, the need to be able to respond to an ever-changing environment is crucial to survive and thrive. Perhaps that has always been the case, but what is different about the 21st century is the speed at which you need to change. The combination of globalization, climate change, and the digital revolution have made life ever-changing, unpredictable, and complex. But the 21st century is not all doom and gloom. We have at our disposal technology that we have never had before—if only we can get out of our own way and use it effectively. Human beings are amazing at solving problems if you give them the right tools, and a clear unobstructed view of the problem they need to solve.

And therein lies the rub. The reality is that organizations are constructed to support a very different paradigm. A paradigm where change is slow and efficiency is more important than effectiveness. Where solving problems is the province of the few. And where trust will never outweigh risk. This paradigm is rooted in the ideas of mass production, a model that has done amazing things but in which flexibility is never the priority. But things can change. For example, Tesla changes its production process so frequently that every car is considered to be unique, requiring automated crash testing and external sign-off for every car. Imagine building a car production line where change is normal, where cars continue to change even when they have left the factory. How does that affect the organization? What do the processes look like? What are the roles and responsibilities? How are risk and compliance handled?

Building an organization for change is difficult enough, but it is made 1000 times harder when you have an existing structure, servicing an existing set of customers, with existing products, all while not forgetting the legacy of existing systems, software, processes, and job structures. Not only do you have to create a new approach, but you also have to create the right environment for change.

Over the last 25 years, I have been lucky to spend time with hundreds of organizations at different stages in this change process. I have seen innovation studios, product factories, and tribes. I have seen companies acquire the change. I have witnessed companies aligned to OKRs, value streams, business processes, customer segments, and product lines. All of these approaches had good and bad elements, and their ultimate success hinged not on the approach but rather on the nature of the people who drove the change and whom the change was affecting. It was neither the approach nor the people alone that were a barometer of success, but instead the combination of those two things.

In this book, Cesário and Ilia have provided a unique view of organizational agility. They first describe the foundational ideas that enable someone to approach the problem. These are the tools and vocabulary needed to think about Agile enterprise-level change. The second section is a practical approach to adoption. It marries the foundational elements into a people- and customer-centric approach to change. The book is epic in scope, and some ideas have to be handled at a high level, but it manages to present a cookbook for Agile adoption and change. In fact, the authors' initial

motivation for the book was driven by the need to bring consultants up to speed on their Agile enterprise change engagements. For some, this book will be a roadmap for future research and learnings; for others, it will serve as a guidebook for their own organization. Whatever your motivation for reading this book, it provides a great resource as we embark on transforming our organizations to thrive in the 21st century.

Good luck with your transformation and enjoy the journey (with this book in hand).

— Dave West, CEO Scrum.org
Boston, Massachusetts

Preface

Welcome to *Creating Agile Organizations*, a book that explains how to guide a successful Agile transformation in your organization.

The term "Agile," as we use it in this book, is defined in the Agile Manifesto, which first appeared in 2001 and defined a set of software development values and principles. The Agile Manifesto was a reaction to the unsuccessful linear development process—the dominant way of working at that time. The people who created the Agile Manifesto all had their own Agile approaches, such as Scrum, DSDM, FDD, Extreme Programming, Adaptive Agile Development, and Crystal, among others, but nonetheless agreed on 5 values and 12 principles.

The popularity of the Agile approach has since grown rapidly across the software development world and beyond. One of the most popular approaches to Agile today is known as Scrum. The Scrum framework was created in 1993—eight years before the Agile Manifesto—and introduced to the broader public in 1995. This book uses the Scrum framework as the foundation to create agility. But we go far beyond the Scrum framework and add the practices and guides needed to become an Agile organization.

Most Agile approaches like Scrum are easy to understand. Actually applying them, though, implies changes to the structure, processes, and policies of an organization.

Once an organization understands and envisions what it wants to become, it is usually still a very long and painful journey to get there. There are many obstacles on the road to agility, including power politics, hidden agendas, resistance, lack of skills and competences, and many more.

Why This Book?

The concepts in this book are based on decades of guiding large- and small-scale Agile adoptions around the world. During those years, we observed many organizations that struggled to scale Agile to bigger product groups. We came to realize that the biggest pitfall is that of local optimization and its implications for the resulting organization design.

In a nutshell, the Agile adoption approach we encounter is this: Organizations typically start with a few Agile teams in a bottom-up manner. As these teams become successful individually, their successes draw the attention of the larger organization. And, as a result, the broader organization wants to apply what works so well for individual teams, with bigger development groups inside and outside software development. Copying what works for a single team to the larger organization is unlikely to succeed. Most attempted adoptions use some level of Scrum at the team level only and leave the rest of the organization design intact. The teams are responsible for a narrow part of the product and focus on increasing their individual team performance. The false assumption is that the performance of the entire group will increase because the separate teams improve their performance. Contrary to what many might think at first, improving the teams independently will often

not improve the group's performance as a whole. The whole group's performance depends on how the teams interact with each other and how they are tuned, not on how they perform separately. What is needed is an approach that optimizes the entire group, which we share with you in this book.

In this book, we share coaching techniques and practices that we applied in successful change initiatives and found helpful to most organizations. We describe how to work at the level of organizational design, senior and middle management, the product group, and the team.

What Makes an Organization "Agile"?

An Agile organization can adapt to changing market demands both rapidly and at a low cost. It aims to learn faster than its competitors and uses that learning to invest in the important work and to avoid wasting resources on low-value work.

Organizational agility comes from the following capabilities:

- Work is completed in short iterations in which ideas are turned into value, each time providing the opportunity to effectively change direction.
- Short feedback loops enable fast learning about the offered products, services, user needs, technology, and internal processes.
- Teams can solve a broad range of user problems, thereby minimizing specialization bottlenecks and improving time-to-market.

At the heart of Agile organizations are self-managing, cross-functional teams that solve end-user needs and create value for the company. In Agile organizations, the teams self-organize to use their full intellectual potential. And, if they organize suboptimally, they correct and continually adjust as they find out more—that is, through bottom-up intelligence.

Yet having many Agile teams is not sufficient for achieving agility at scale—you also need the right organizational design that enables those teams to be successful. Many traditional organizational designs based on the principles of Taylorism optimize more for control, predictability, and resource utilization than for adaptability and learning. Delayed learning makes it difficult to decide on the right product and process improvements, and rigidity makes reacting to new insights costly. Management faces the task of redesigning the organization so that agility at scale becomes possible.

Typically, this means:

- Organizing teams around products to optimize for satisfying customer needs
- Developing groups that can work across the whole product value stream to meet business objectives and market needs
- Moving to a management system that values adaptability and learning over control and predictability

Coaching organizations for adaptability starts with understanding the impact an Agile setup has on the current organizational design. Once an organization understands and accepts the design implications, the next step is to discover and develop an organizational structure that works in its specific context. People who create their own process will feel ownership of it, and therefore are more likely to evolve and improve it over time.

Intended Audience

We wrote this book with several audiences in mind. First are Scrum Masters and Agile coaches who guide teams and organizations in Agile adoptions. To them, this book offers not only practical coaching techniques but also deep insights into organizational dynamics and design implications.

We also wrote this book for managers who want to scale agility to their broader organization. They can change the organizational dynamics and interactions required for adaptability by redesigning organizational structure, policies, processes, human operations, and management behavior.

In general, this book will be helpful for anyone looking to improve their current Agile setup.

We hope that you, the reader of this book, will enjoy the benefits of *Creating Agile Organizations* and will be inspired to face your organizational design challenges head-on.

—Cesário A. Oliveira Ramos
Hengelo, The Netherlands

—Ilia Pavlichenko
Istanbul, Turkey

This page intentionally left blank

Introduction

This book is about transforming an organization into one that can respond effectively to market changes and delight the customer. It is about redesigning the organization into an Agile one as effectively and painlessly as possible to create continuous innovation and customer value.

Bringing agility into an organization is hard work, and it requires patience, humor, and long-term commitment. It can take many years to simplify your organization and discover how to make agility work in your organization. But once you have been through all that, you ought to be able to react more effectively to changing market conditions. Unfortunately, there is no magic wand you can wave to complete this transformation, nor is there an oracle you can consult who has all the answers to your problems. Instead, you must find *your* answers and discover your organizational model one step at a time.

Agility at the Organization Level

Many problems at the workplace are not the fault of the individual managers or teams. Instead, they are often the result of the organizational design in which those managers and teams work. According to Jay Galbraith, an organizational design consists of five major interrelated components[1]:

- **Strategy**: The primary direction of the company. The goals and objectives, values and mission, products or services to be provided, markets to be served, value offered, and sources of competitive advantage.
- **Structures**: The units, roles, responsibilities, and relationships.
- **Processes**: The flow of information, both horizontal (workflows, product development, value delivery) and vertical (business planning, budgeting), as well as the coordination between units.
- **Reward systems**: The alignment of employee goals with the goals of organization through salaries, promotions, bonuses, profit sharing, stock options, recognition, and challenging assignments.
- **People practices**: How to develop, rotate, and recruit people. Which capabilities and mindsets are needed.

Galbraith created the Star Model for thinking holistically about these components. When all components are aligned and reinforce each other, the organization is most effective because people get a consistent message about the appropriate behavior. In an Agile organization, this component alignment means that the organizational design supports—rather than limits—the people's ability to effectively adapt to customer demands.

> For an organization to be effective, all the policies must be aligned and interacting harmoniously with one another.[2]

To implement this strategy, a company requires certain capabilities. These capabilities can be developed by ensuring the organization has the right processes, structures, people practices, and reward system in place. The reward system should be congruent with the structure and processes to influence the strategic direction. The challenge is to design each of these components into a system of work that allows the required Agile capabilities to be developed over time.

A Systems Thinking Approach to Adopting Agility

Why do people repeat the same unsuccessful behavior in organizations? Why do certain problems in organizations seem to appear again and again? Why aren't improvement actions leading to the expected results? What is probably going on is that the improvement actions are anti-systemic, acting on independent problem events without paying close attention to their relationships and deeper causes.

An organizational design includes, among other things, decisions about the division of work and the assignment of responsibility among units. When units coordinate to get the job done, they create complex interactions and feedback loops—the system's structure—that generate the observable recurring problems. You address these problems by understanding the system's structure and then redesigning the organization to create new interactions at the workplace.

> A classic example of system effects: We observe a slight decrease in productivity in a software development team and the team gets behind schedule. As a quick fix, project management pressures the developers to step up and "just make it happen." This quick fix solves the problem for the upcoming release because the developers take shortcuts and lower quality to meet the schedule. For the next release, the team works with a lower-quality system, so their productivity is worse than before owing to the accumulation of the technical debt. What will likely happen is project management pressuring the development team again, worsening the situation. Both parties behave "rationally" from their point of view with good intentions.

When you see how the system of work constrains people's behavior, you realize that the recurring problems are mainly due to the system, not to the people themselves. Thus, working on the people when the problem is actually systemic in nature is not the right path to go down. You can achieve minor improvements at best.

> Continuing the preceding example: Release after release, management notices that the product quality is decreasing over time; there are many bugs and multiple user complaints. They decide to send the team to train on software quality. This action will likely help only in the very short term. Why? Because project management still pressures the team to meet the schedule. Eventually, the team is forced to cut corners again, even though they know how to write quality software: The system's dynamics are left intact and generate the same behavior patterns.

The system of work determines, to a large extent, how people cooperate, what the prevailing mental models are, and how work gets done. It also determines what is valued, which leadership style is employed, and even which color paint is used on the walls. To change all this takes time and requires

that people unlearn old lessons and relearn new ones. The current organizational design decisions result from how the designers think the world works—their mental models. Thus, in Agile adoption, it's important to help the organization become aware of how the current system of work supports or obstructs achieving their agility goals. It can then use that understanding to make improvements to its systems.

The Importance of Self-Organization for Agility

The basis for agility is an environment of exploring new possibilities for improvement. Such an environment cannot overly constrain people's exploration with detailed procedures and rules because that reduces the space for innovative ideas—precisely the opposite of what is needed. Instead, management needs to support self-organization by establishing a few simple rules, a clear goal, and short feedback loops.

A classic example that shows how self-organization works is a flock of birds. How do they keep flying in the V-formation while they change direction and reach their goal? The answer lies in self-organization based on simple rules and frequent feedback. Although the flock of birds seems centrally organized, the individual behavior of the birds is not. Each bird organizes itself so that it does not lose sight of the others but also does not bump into the others. If we had to simulate this, we would have some simple rules. One of those rules could be something like: Always keep a distance between 30 and 50 cm from each other. The emergent behavior of the flock is the result. If we try to control the V-formation by managing the individual behavior of each bird, we will not be effective. There are too many interactions happening to be able to understand what is going on. Moreover, the emergent V-formation cannot be understood by studying each bird individually because it emerges from the interactions and feedback loops.

In our organizational context, by necessity, no single person can understand or know everything about the transformation and solve all the problems. You cannot have one person telling a large group of people exactly what to do. But you also cannot have a large group of people doing whatever they want when they want to. What you *can* do is provide clear focus and direction, ensure that people take responsibility for solving the problems that they own, and provide the support they deserve. Such an approach provides control and makes better use of the individuals' intellect, but also simultaneously gives them flexibility to explore solutions.

A New Role for Management

The role of operational management is likely to change. From our experience, many Agile organizations end up with fewer operational management roles. Why? Because a lot of the operational work becomes the responsibility of the teams. For example, a product person now makes decisions about the product, and the self-managing teams take care of coordination and decide how to do their work. Therefore, operational managers no longer need to divide the work, decide what people should do, or coordinate between people or teams. Instead, they can devote their whole intellect to making it possible for the teams to be successful. Improving the organizational system of work and mentoring people on problem solving is the new reality for them.

A Learning Organization

A large part of Agile adoption focuses on learning how to work with self-managing teams in short iterations of learning. People need to learn new things—for example, new work practices, teamwork, and ways to lead Agile teams. Also, the decision-making process, communication structures, and measures of progress are subject to change. A second part of Agile adoption is the change to a mindset of continual improvement. This covers a broad spectrum of people—from R&D, business, and management to sales, marketing, and other disciplines.

There are roughly two ways to begin the transformation: either the organization is starting from scratch, or it has already been working with Agile for some time. In the former case, you have the benefit that you can start with a greenfield. A downside is that you are at the start line, and a long bumpy road lies ahead. But the latter case might be even worse! The organization might have been doing Agile for a long time—just doing it wrong and not getting the expected benefits. You will need to repair misunderstandings and facilitate relearning to help it take the next step.

When people immerse themselves in a new experience and uncover better ways of doing their work, the new experiences influence a person's mental model about how things work. Gradually new practices become "the way we do things here," and new patterns of behaviors emerge in the organization.

Ownership for Continual Improvement

In an article published in *The Journal of Personality and Social Psychology*,[3] Ellen J. Langer of Yale University described an experiment with lottery tickets to illustrate the phenomenon known as the illusion of control. Two groups were involved in this experiment. The people in the first group received lottery tickets that were preselected for them, while the people in the second group got to choose their tickets and numbers. When they had their tickets, the experimenter asked people in both groups to sell back their tickets. As predicted, the people who got to choose their tickets wanted to receive a higher amount of money than the ones who got a ticket assigned. Why does this happen? To quote the paper:

> [W]hen a chance situation mimics a skill situation, people behave as if they have control over the uncontrollable event even when the fact that success or failure depends on chance is salient.[3]

The people who chose their tickets also felt more ownership of them, which is an essential observation for us. When you create something yourself, you feel ownership of it, and once you own it, you might care enough to improve it. You can say the same thing about teams that create their work process: They feel ownership of their approach, which makes them take responsibility for improving it. Once teams discover and learn what works for them, they can share that learning with other teams across the organization. This process of lateral learning enables the organization to discover which changes it needs to make to build a model that works in its context.

Empirical Process Control

Successfully bringing agility into an organization is a journey of discovery and learning. There is no fixed end-state when you can say, "The transformation is done." Instead, the end-state is a dynamic one, in which new insights are continually used to improve the product, processes, and organizational design.

An organization needs to change itself to become Agile. The challenge is that one cannot know upfront exactly what needs to change or precisely how the future state should look; it is a complex problem of innovation. Such a problem cannot be solved by centralized control, imposed order, and prediction. Instead, you must use an empirical process in which you take an informed step forward, inspect the results, and then adapt your plan based on what was learned. This approach is not only useful for the adoption of an Agile organizational design, but also an essential part of running an Agile business.

The Scrum Framework

The Scrum[4] framework enables empiricism through transparency, inspection, and adaptation—and that is why we use the Scrum framework as the basis for the transformation process. Scrum is a framework that you can use to build your process; it is not a framework that builds your product. Methods give a sequence of steps to build a product—they are a procedure for accomplishing something—and tell you how to do things. However, Scrum makes you aware of problems that are preventing you from reaching your goals. Once these problems become clear, you have the opportunity to do something about them.

How to Start?

Having an inspiring or urgent reason to change is a precondition for a successful transformation. Without it, the chances of lasting change are slim. After all, why would we change if all is running perfectly? Once the reason is there, then the hard work begins.

A Quick Tour of the Book

This book shares how we applied many research findings to our Agile transformations as effectively and painlessly as possible. Readers will find the book most valuable if they have a good understanding of the Scrum framework. We don't intend for this book to be an introduction to Scrum.

The book can be roughly divided into two parts. The first part, *Foundational Concepts*, includes systems thinking, basics of flow and resource efficiency, guidelines for organizational design, preferred coaching approach, and guidelines for productive change. The second half of the book, *Applying the Concepts*, offers practical tools for large-scale organizational Agile adoption—for example, defining a product workshop, tools for preparing and facilitating large-scale Agile events, and guides for working with teams and leadership. The tools are explained using various real-life examples of organizations, Product Owners, and teams on their journey to becoming an Agile company.

The examples show how to prepare, structure, and guide large-scale Agile adoptions. They provide real-life examples of organizational designs, challenges, solutions, and pitfalls that you can learn from.

Finally, we share a few case studies that illustrate how Scrum can be used to apply agility successfully at a large scale.

Part I: Foundational Concepts

Chapter 1: Organizing for Adaptability focuses on the problems that organizations face when adopting agility at scale. It describes typical problems, organizational design flaws, and what must change if the organization is to be successful.

Chapter 2: Systems Thinking describes the overarching approach to improve on organization. We introduce the basic concepts and techniques and lay the foundation for the rest of the book.

Chapter 3: Optimize for Adaptiveness is about the strategic choices regarding optimization goals. It provides the basics of lean thinking and queueing theory to enable managers to make informed trade-offs between flow and resource efficiency.

Chapter 4: Agile Organizational Design describes organizational design principles for use when working at the leadership level of an organization.

Chapter 5: An Agile Adoption Approach provides an overview of the adoption approach. We share a bird's-eye view of the activities we found crucial for a large-scale adoption.

Chapter 6: Coaching for Change provides guiding principles on how to make the change successful in your organization.

Part II: Applying the Concepts

Chapter 7: Group Facilitation covers the many occasions in which you need to facilitate a group of people. In this chapter, we share how to prepare, design, and facilitate workshops.

Chapter 8: Preparing the Product Group describes how to do just enough preparation to successfully start your transformation. We explain how to align management around their objectives, identify what needs to change, define your product, and identify the organizational elements to form the product group.

Chapter 9: Launching the Product Group focuses on how to successfully launch the product group and prepare to work in the new organizational design. We provide a set of activities and techniques you can consider in your launch.

Chapter 10: Coaching Teams is about working with teams in a large group. We delve into what teams are and what you as a coach can do to guide team development. We discuss techniques you can use for coaching, and explore productive team relationships.

Chapter 11: Guiding the Product Ownership focuses on ways to create and service product ownership within a large product group. Who is the right person to lead the group? How you can move away from output and to outcome-based measures and how you can work with multiple teams are other topics covered in this chapter.

The book concludes with the **Appendix: Workshop Examples**, which highlight real-life examples to guide you on your own Agile transformation journey.

It is our hope that you, the reader of this book, will enjoy the benefits of *Creating Agile Organizations*, and be inspired to discover your path to agility.

References

1. Jay Galbraith et al. *Designing Dynamic Organizations: A Hands-on Guide for Leaders at All Levels* (Amacom, 2001).
2. Jay R. Galbraith. "Star Model." www.jaygalbraith.com/services/star-model.
3. Ellen J. Langer. "The Illusion of Control." *Journal of Personality and Social Psychology* 32, no. 2 (1975): 311–328. https://doi.org/10.1037/0022-3514.32.2.311.
4. "The 2020 Scrum Guide." https://scrumguides.org/scrum-guide.htm.

This page intentionally left blank

Acknowledgments

A very special thanks to our manuscript reviewers, who worked through the early versions of the manuscript to give us feedback. Thanks to Rick Ijspeerd, Craig Larman, Martin Mandischer, Hannes Eickmann, Annebeth van Hall, Elisabeth Geyer-Schall, Ademar Aguiar, Roman Lobus, Vincent Wensink, Arjen Uittenbogaard, Leise Astrid Passer Jensen, Denis Sunny, Venkatesh N. Krishnamurthy, Simon Reindl, Pascal Dufour, and Jowen Mei for their all their effort to point out mistakes, vague sections, missing parts, plain old errors, and opportunities for improvement.

We want to make a special mention of Jim Coplien, who spent many days providing detailed feedback. His brilliant and detailed comments, concrete suggestions, and provocative statements made us critically examine our manuscript and helped us dramatically improve its structure and content. Many of his remarks made us laugh and cry simultaneously; we want to share two of such comments:

> "This is not an experiment. It's just let's-fart-around-and-see-what-happens."

> "Again, this can be the topic of a whole university degree. Thank heavens you have dedicated a whole chapter to it. I will really feel prepared—almost as much as if I had a coaching certificate."

Thank you, Sensei, for being so hard on us; we humbly appreciate it!

Many thanks to Dave West and Kurt Bittner, who got us started writing this book. Our warmest thanks go to our editor from Pearson, Menka Mehta, who worked patiently with us through many missed deadlines to get this book to done.

And last, many thanks to our graphical artist Arqam ul Hassan for turning our sketches into outstanding graphics.

This page intentionally left blank

About the Authors

Cesário Oliveira Ramos works worldwide as a senior management consultant on large-scale Agile adoptions in the financial and high-tech industries. His international experience, strong background in technology, and passion for people make him an influential partner in organization design and in leading the adoption of Agile. Cesário has a MSc in mathematics and computing science from the University of Eindhoven and is also a Certified LeSS (Large-Scale Scrum) Trainer and Professional Scrum Trainer from Scrum.org. Outside of his consulting work, Cesário has served as a Java/C# developer, lead software architect, CTO, and product manager. He is also the author of *Create Agile Organizations, A Scrum Book,* and *EMERGENT: Lean & Agile Adoption for an Innovative Workplace*.

Ilia Pavlichenko leads Agile transformations in companies around the world. He is an organization design consultant with a rigorous focus on coaching senior management. Ilia regularly speaks at international Agile/Scrum conferences. Ilia is also a Certified LeSS (Large-Scale Scrum) Trainer and a Professional Scrum Trainer from Scrum.org. In the past he was a programmer, project manager, Scrum Master, and Agile Coach at multiple product organizations. Ilia is a candidate for Master of Sports in table tennis and streetlifting, and is a fan of Queen and Brian May.

This page intentionally left blank

PART I

FOUNDATIONAL CONCEPTS

This page intentionally left blank

1

Organizing for Adaptability

Agility is the ability of an organization to adapt to new conditions and to change its direction while creating maximum value and customer experience.
—Mike Beedle

The basic building blocks of an Agile organization are self-managing cross-functional teams that can create customer value. When you combine many such teams, you can create larger groups that self-organize effectively around the ever-changing market needs. To make that possible, you need to understand the typical problems and pitfalls that organizations face when adopting agility at scale, and what to change so that your organization can be successful.

Hello, VUCA World!

We live in exponential times—that is, a time when the speed of change is increasing exponentially. VUCA is an acronym used to describe or reflect on the volatility, uncertainty, complexity, and ambiguity of general conditions and situations.

Here are just a few facts to get us started:

- A business cycle that was 75 years long 50 years ago was only 7 years in 2017.
- As many as 93% of U.S. multinational enterprises are changing their business models.[1]
- A higher "percentage of profit and Revenue [have come] from NEW products and services in the last five years"—50% on average, and 70% in more competitive industries.[2]
- Half of the companies in the *Fortune* 500 have disappeared since the year 2000 (according to Pierre Nanteme, CEO of Accenture).

To cope with this fast-changing environment, it is not a surprise that organizations are striving for agility these days. Indeed, many organizations are now asking this question: How can we become sufficiently Agile to reach our goals? Although agility might help the organization to achieve some of those goals, it is not a silver bullet, and "responding to change" is not enough. The Agile approach is to work in customer-driven iterations that frequently deliver value to the end user or customer. Adopting Agile at scale requires a supporting organization design.

What Is an Agile Organization Design?

An organization has a design, just like a car. Likewise, software systems have designs, too. Each design has specific optimization goals it intends to achieve. For example, a key optimization goal when Toyota was developing its Prius car model was fuel efficiency and aerodynamics—but not maneuverability at top speed. You could try to win a Formula 1 race with a Prius, but that probably won't work.

The architecture that Amazon.com uses has as an optimization goal to be able to build many software components rapidly and independently. To make this possible, the company uses a service-oriented architecture.[3]

Another example is the Formula 1 pitstop team: Their configuration and process enable them to change the tires and refuel a Formula 1 car in less than 2 seconds (Figure 1.1).

Figure 1.1
Formula 1 pitstop team.

Many people are needed to achieve that goal. If you look closely at a pitstop in action, you will realize that each person spends most of the time waiting on others. You can observe that one person waits for the wheel nut to be removed so that he can then remove the old tire. Another person waits for the old tire to be removed so that he can then place the new tire; meanwhile, the person who removed the wheel nut is waiting for the new tire to be placed so that he can tighten the wheel nut again. The Formula 1 pitstop team process and the car it services are designed for speed, not for resource efficiency. But what happens if you bring a Formula 1 pitstop team and process to a big truck and ask them to change four tires in less than 2 seconds? You can try, but that probably will not work because the truck's design is not optimized for doing that; instead, it is optimized to meet other goals.

What Does Your Organization Optimize For?

Organizational designs have optimizing goals, too. An organizational design is a combination of roles, responsibilities, reward systems, coordination and reporting lines, structures, processes, and policies intended to execute the business strategy. A few examples of possible optimizing goals could be the following:

- **Resource efficiency:** Efficient use of people and other resources.
- **Ideation:** Maximize the number of innovative ideas and creativity.
- **Learning:** Gain knowledge about customers and their needs, the market, and new technology.
- **Security:** Critical information protection.
- **Speed:** Shortest time from idea to delivery.
- **Adaptability:** Ability for an organization to change direction fast and at a low cost.
- **Reliability:** Delivery with low deviations from the plan.
- **Safety:** Protection from danger, risk, or injury.
- **Quality:** Conformance to standards, product excellence.

Senior managers are organizational designers. They should pick a few optimization goals they believe are crucial for their organization to pursue so that it can reach its business objectives and communicate those goals to the whole organization. Alignment around optimization goals facilitates decision making and reduces ambiguity. In such a case, many management decisions about organizational design can be verified to see if they are in line with the chosen optimization goals.

Optimization Goals and Indirect Wishes

An organization should be aware of the difference between its optimization goals and its indirect wishes or business goals. Here are a few examples of the latter: reduce operational costs, increase revenue, or increase market share. Indirect desires are not influenced by the system of work directly, but rather are the intended outcome of working in such a system. For example, organizations often

optimize for agility because they assume it will help them achieve their business objectives; agility itself is not the business objective, but rather the means to reach it. When an organization wants to optimize for agility, it needs a proper organizational design for that. But what does it mean to optimize for agility?

Optimizing Goals of an Agile Organization

According to Mike Beedle, agility is the capability of an organization to adapt to new conditions and to change its direction while creating maximum value and customer experience. We love this definition. This definition impacts an organization at two different levels. First, it addresses agility at the overall organizational level:

> *Flexibility is the ability and capacity to reposition resources and functions of the organisation in a manner consistent with the evolving strategy of management as they respond, proactively or reactively, to change in the environment.*[4]

Such flexibility is useful for reacting to opportunities by starting, combining, or ending products or services in an organization.

The second level is about the adaptability of products or services themselves and addresses the following question:

> *How can we effectively adapt our product to changing market demands fast to cope with changing user needs?*

An Agile organization can do both; however, in this book, we focus on the latter. An Agile organization is able to switch between working on one job to another both quickly and at low cost.

Changing direction is key, but you must also decide in which direction to go. How do you know in which direction to adapt? You need to have a feedback loop. Short feedback loops increase learning, so if you want to be Agile, you need an organizational design that optimizes for the following:

- **Learning about the value stream:** Learning quickly about users' needs, product features, and market trends. Speed of delivery is necessary but not sufficient. After shipment, you still need to gather feedback from the market, reflect on it, and act on it.
- **Learning about the product organization:** Having short feedback loops for the processes used, organizational design, technologies, and team capabilities.
- **Adaptability:** Being able to use both types of learning to change direction at the level of the product group by switching between jobs quickly and at a low cost while creating maximum value.

Typical Problems When Adopting Agility

It is relatively simple to achieve agility with one team that has all the needed competencies and technical skills; however, many larger organizations usually have multiple teams that need to work together on a single value proposition or product. They need to scale Agile across all those teams so that they can become adaptable at scale. But scaling what works so well for individual teams

working separately to larger groups working together turns out to be a different and more difficult problem to solve. While each separate team might be Agile locally, that doesn't mean the group will also be Agile. An important reason for this difference is that the more teams there are, the more likely it becomes that they will specialize in a single function, skill, or technology—for example, product management teams, marketing teams, IT teams, development teams, or manufacturing teams. In an organization with such a structure, interdependencies typically exist between the teams. The unfinished product moves from one group to the other, with each group adding something before the product is delivered to the customer.

Specializing the teams facilitates integration within the specialization but complicates integration across the groups. Why? The teams are responsible for completing their part and then handing the product over to the next unit; therefore, they do not feel the need to communicate frequently with other teams. When they do communicate, they might even have difficulty understanding the other teams' perspectives. Numerous problems can follow from that. For example:

- A lack of customer and market information from the business teams makes it difficult for development teams to deliver successful products.
- The development teams might not necessarily focus on the requirements that the business department believes are the most important.
- The development teams might ignore requirements regarding manufacturability that increase manufacturing time and cost.

The more the teams specialize, the more likely it becomes that they cannot deliver end-customer value, but only provide a part of the desired value. All these parts need to be identified, planned, integrated, and coordinated to yield the complete end-to-end value before an organization can comprehensively understand what is going on. Much of this activity is unnecessary and delays learning and, therefore, the ability to adapt in the right direction.

Systems Thinking theory focuses on the complete picture to improve the end-to-end process; it involves studying the broader system behavior over time and using that understanding to improve. The important insight from Systems Thinking theory is that the performance of an interdependent group of teams depends primarily on how the teams are tuned, not on individual team autonomy or how the teams perform as separate entities. In Chapter 2, "Systems Thinking," we cover in more detail how to use this approach in Agile organizations.

The Functional Hierarchy Organization Design

Typical organization design is the functional hierarchy. As Peter Scholtes described in *The Leaders' Handbook*,[5] a severe accident between passenger trains in the year 1841 heavily influenced how we design our organizations today. According to Scholtes, the railroad company wanted to ensure that such an accident could not happen again. A decision had to be made about how to reorganize the railroad management system. The railroad had a choice between the two most well-known organization designs at the time—the military and the church. The military was a top-down hierarchical structure, whereas the church used a distributed structure. After careful consideration, the railroad opted for the top-down hierarchy because it optimized for control and enabled the company to find the cause of problems quickly. Its decision still influences current-day organizations, as illustrated in Figure 1.2.

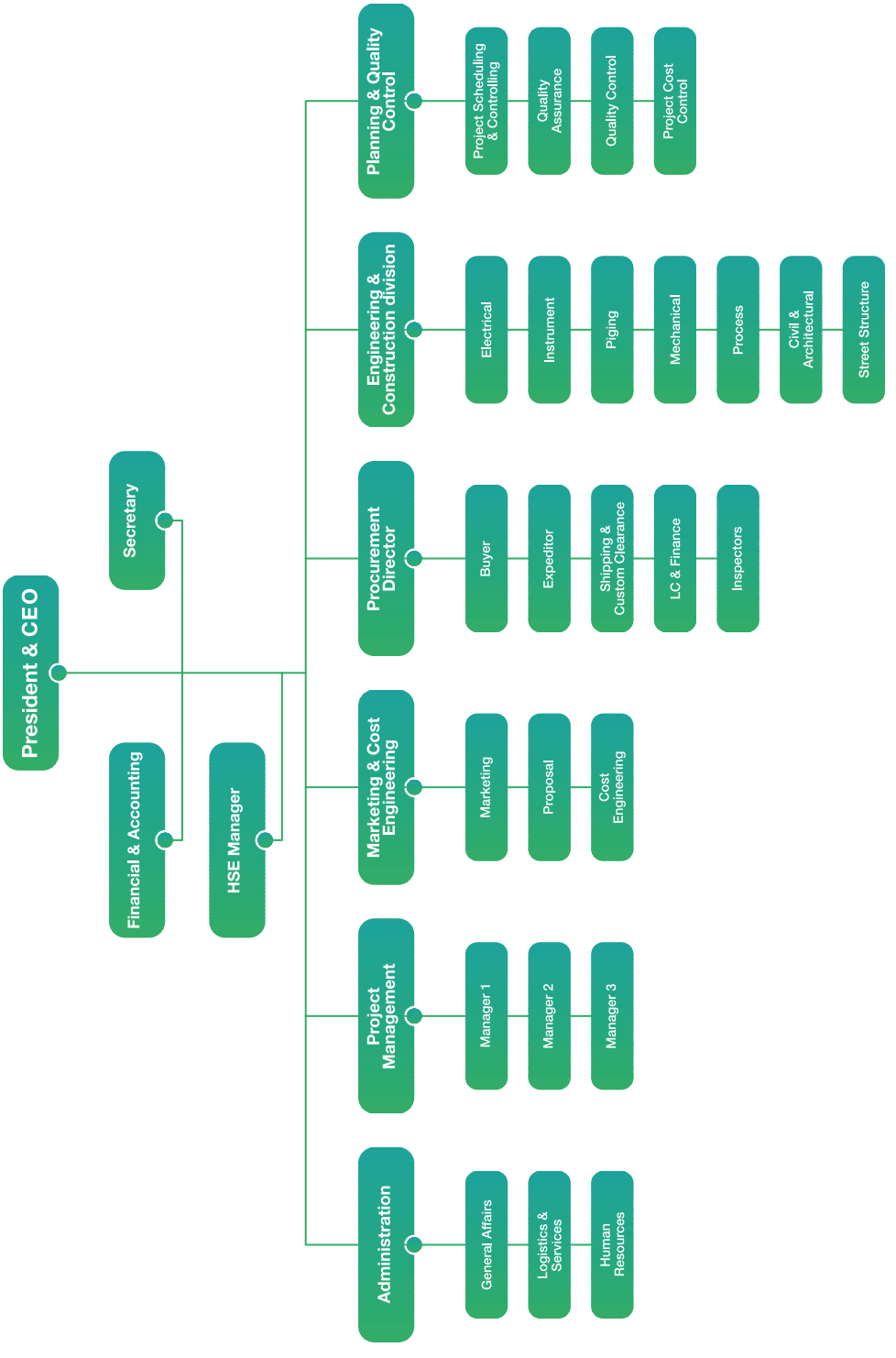

Figure 1.2
Hierarchy organization design.

Specializing the Work

In the first half of the 1900s, Henry Ford and Frederick Winslow Taylor heavily influenced management thinking. Ford introduced the very successful automobile production line, while Taylor focused on his scientific management approach. Both liked to specialize the work and divide work into separate functional tasks. Narrow functional tasks enabled people to concentrate on doing a simple job, so that people with poor or no education could work efficiently.

Separate the Head from the Hand

Taylor also introduced the concept of *separating the head from the hands*. Educated people would design the process and partition the work into dumbed-down tasks, and then people could be attached to those tasks. Furthermore, Taylor stated that there was a best way of doing the work and that management's job was to find that one best way and then let the workers do it. Along with the best way came measures, and later key performance indicators, and standard times for each of the tasks.

Ford had a similarly dim view of workers' capabilities, complaining, "Why is it every time I ask for a pair of hands, they come with a brain attached?" In Ford's system, people were only as valuable as the simple, repetitive tasks that they could perform. In essence, they were viewed as "interchangeable, nameless, and faceless."

Fast Forward to Present Times

In organizations that do large-scale development, much of the thinking advocated by Taylor and Ford still prevails today. These organizations are designed as a series of functional groups, with the aim being that each of the groups work individually at maximum efficiency. The resulting silos are managed by single-function managers and have measures—key performance indicators—to assess how well they are performing. Figure 1.3 illustrates a typical example of such an organizational design in the context of software development.

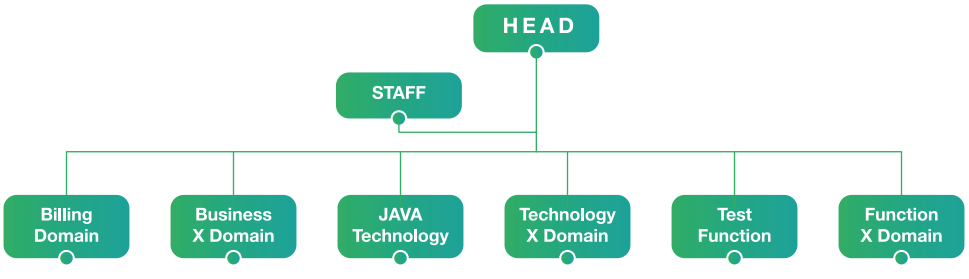

Figure 1.3
Typical silo organizational design in modern times.

You can see domain groups that specialize in a part of the business—for example, billing. Next to that, you can see specialization in technology such as Java, as well as specialization along with functions such as testing or marketing. There is nothing wrong with this kind of functional hierarchy as long as its optimizing goals are in line with your organization's goals and it helps you in reaching your business objectives. John Kotter, Professor of Leadership, Emeritus, at the Harvard Business School, highlights some important goals of this kind of hierarchy:

> *[A]t both a philosophical and a practical level, the Hierarchy (with its management processes) opposes change. It strives to eliminate anomaly, standardize processes, solve short-term problems, and achieve stopwatch efficiency within its current mode of operating.*[6]

The Agile optimizing goals are not in line with the purpose of the functional hierarchy. This organizational design invites managers to optimize the silos separately, but customer value flows horizontally across silos—not vertically across the hierarchy. So, if you want the organization to learn fast and use that learning to correct its direction, you need to focus on how the silos interact, not on how they perform separately. This implies minimizing costs when switching between jobs and getting the work through your organization as effectively as required. But how do you do that? By minimizing task switching costs and transitioning from resource efficiency to flow efficiency. In the context of software development, for resource efficiency it is more important to ensure each team always has a feature to work on. For flow efficiency, it is more important that a feature is always being worked on. So, in an organization focused on resource efficiency, the work is likely queued before each team with the goal of always keeping them busy. In contrast, in an organization that emphasizes flow efficiency, the goal is for teams to always be ready to pick up work, which implies that teams are expected to learn to understand and work effectively on multiple topics; if there is nothing to work on, then the teams must be idle sometimes.

An excellent way to achieve agility is by redesigning your organization. If you are not willing to make this effort, then scaling Agile in an organization with a functional hierarchy design will get you into trouble quickly, as we will show next.

Avoid Copy–Paste Scaling: A Typical Scaling Approach

Scaling is about increasing in size. Fire departments, for example, scale their operations depending on the severity of the fire. Depending on the scenario, they may increase the size of the trucks, the number of vehicles, the number of people, and the coordination and communication process as needed. This approach is what Cesário Ramos calls *copy–paste scaling*.[7] You "copy" the trucks and people needed and "paste" them to form a larger group while adding extra processes for communication and coordination.

When you apply this approach to Agile, it means increasing capacity by copying and pasting Agile teams in your development group. To support and coordinate this growth, organizations typically augment their teams with special roles such as *project managers* or *feature owners*. They also add extra layers of coordination, such as *release management*; extra process steps, such as *integration test phase*; and even additional artifacts, such as *system specifications* and *team work packages*.

Unfortunately, this results in reduced team–customer collaboration because the teams start to focus on the added coordination roles, the intermediate artifacts, and the revised processes. As a result, you are now decreasing the feedback loops and slowing down rather than speeding up.

Let us provide an example.

> One of our clients operates in the energy trading business and has a distributed development group across three sites. It initially started with a few Scrum teams, but quickly scaled up to 16 teams due to market demands over a couple of months.
>
> The development group supports a business process that consists of several steps and technical components, including User Interface, Database, and Workflow. aturally, following a copy–paste scaling approach, they formed their Scrum teams around the steps and components. Each Scrum team had its team Product Backlog, Scrum Master, and team Product Owner. To serve the teams, they had to divide each feature into separate parts corresponding to their specialty. Hence, each team could only deliver a feature part, even though feature delivery mainly required work across multiple business process steps and components. As a result, the teams needed extensive planning and coordination to try to align their work and deliver an integrated product.
>
> In this organizational design, the teams delivered their feature parts out of sync. Why? Because the work that each needed to do for a feature part was not equal. For example, a feature might require a lot of work on the User Interface and very little work on the Database. This imbalance in teamwork meant delayed testing and customer validation until all teams completed their part. Hence, this approach led to long feedback loops and opaque measures of progress. The result was low productivity, high defect rates, and unhappy customers.

Overview of an Agile Organization Design

Many Agile teams use the Scrum framework to get their work done. Scrum was first presented to the broader audience at the 1995 OOPSLA conference as an enhancement of the iterative and incremental approach to software development. Just like many other Agile approaches, Scrum was designed to work with small self-managing cross-functional teams; initially, those teams were defined as including no more than six people.[8]

A Scrum team works in short cycles called *Sprints* that enable feedback about progress toward a goal:

> *Scrum is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems.*[9] —**The 2020 Scrum Guide**

In this book, we use the Agile team as the basic building block of an Agile organization. When you combine many such teams, you can create larger Agile groups that solve complex problems in short cycles.

What Is an Agile Team?

In the book *Creating Effective Teams*, the authors distinguished between a group of people and a team:

> *A work group is composed of members who are striving to create a shared view of goals and to develop an efficient and effective organizational structure in which to accomplish those goals. A work group becomes a team when shared goals have been established and effective methods to accomplish those goals are in place.*[10]

A very important characteristic of Agile teams is that they feel collective ownership of their work and problems. In such a team, its members work on shared goals, and tasks apply to the whole team rather than to individuals. The members also understand and accept their team roles and regularly put team goals above their personal work goals.

The definition of a team provided by Peter G. Northouse, a professor emeritus at Western Michigan University, provides us with a description that applies closely to a team in Scrum:

> *A team is a type of organizational group that is composed of members who are interdependent, who share common goals, and who must coordinate their activities to accomplish these goals.*[11]

Just as on a Scrum team, the members of an Agile team are interdependent and must coordinate to get results. One can expect to find trust, an environment of psychological safety, and confidence among team members regarding their capabilities. Everyone in the team participates in decision making and owns the team decisions; decisions are made based on knowledge, not on hierarchical position.

Another definition provided by Jon R. Katzenbach, highlights the fact that team members have complementary skills and hold themselves accountable:

> *A team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable.*[12]

Although each of these definitions is slightly different, they agree that a team is a group of people who depend on each other's skills to accomplish shared goals. The critical point is that team members do not have an independent job to do, but rather must work together to produce some value—like a product.

How do these small teams work together at scale? We will answer these questions throughout the entirety of this book. For now, we start with an overview of the organizational characteristics you can expect to find in any Agile organizational design.

Structural Characteristics

One can consider a single Scrum team as a small product organization and value stream design. The team design is such that it has all the competencies required to turn its work into increments of customer value delivered in short cycles. You can give the most valuable feature—a complex problem—to such a team, and they can independently deliver a solution into the customer's hands. Working with such a team offers various Agile benefits to an organization, including the following:

- The team can always work on the most valuable work.
- Team independence decreases the time to deliver an increment of value—that is, it increases flow.
- Working in short cycles allows for the possibility of changing direction.
- Team autonomy promotes ownership of team results and team processes.
- Working directly on an end-user problem improves understanding of the customer problem domain.

How can you get the same benefits when you have many teams working on a common goal? When each team can independently deliver the most valuable work into the hands of the customer at least every cycle. We call such a team a *feature team*[13] and consider it to be a key element of an Agile organization (Figure 1.4).

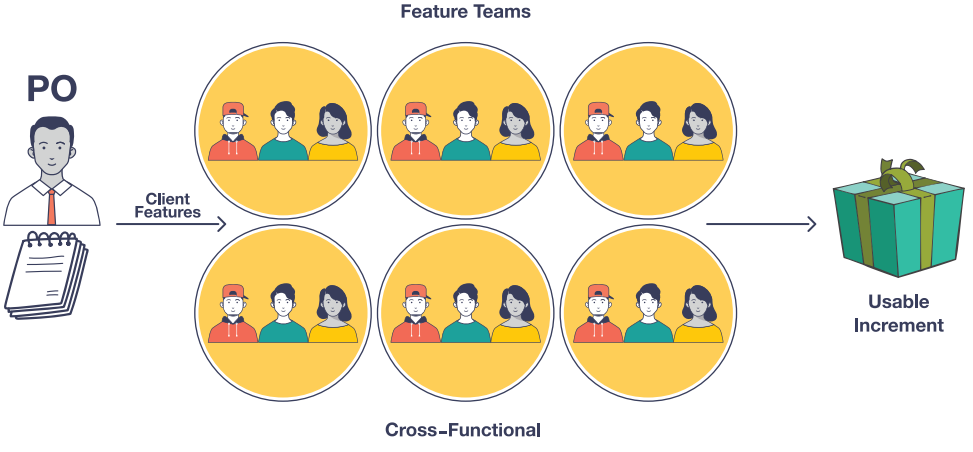

Figure 1.4
Feature teams.

Agility at Scale

A standard view is that scaling agility represents a change for teams only. Therefore, the system of management, the organizational structure, and the policies remain the same. Working in an

organizational design like the functional hierarchy, however, makes it almost impossible to achieve agility at scale. Therefore, management faces the task of redesigning the organization so that agility at scale becomes possible. Typically, this means:

- Organizing a product group around your product or services to optimize for solving customer needs.
- Developing teams that can deliver end-user value that meet business objectives and market needs.
- Moving to a management system that values transparency, learning, and adapting to new realities over detailed planning and measuring performance against them.

What would such an Agile organization look like? The prototype of this organization would have a few shared functions along with the strategic management. The rest of the organization is mainly organized around semi-independent product groups, as illustrated in Figure 1.5.

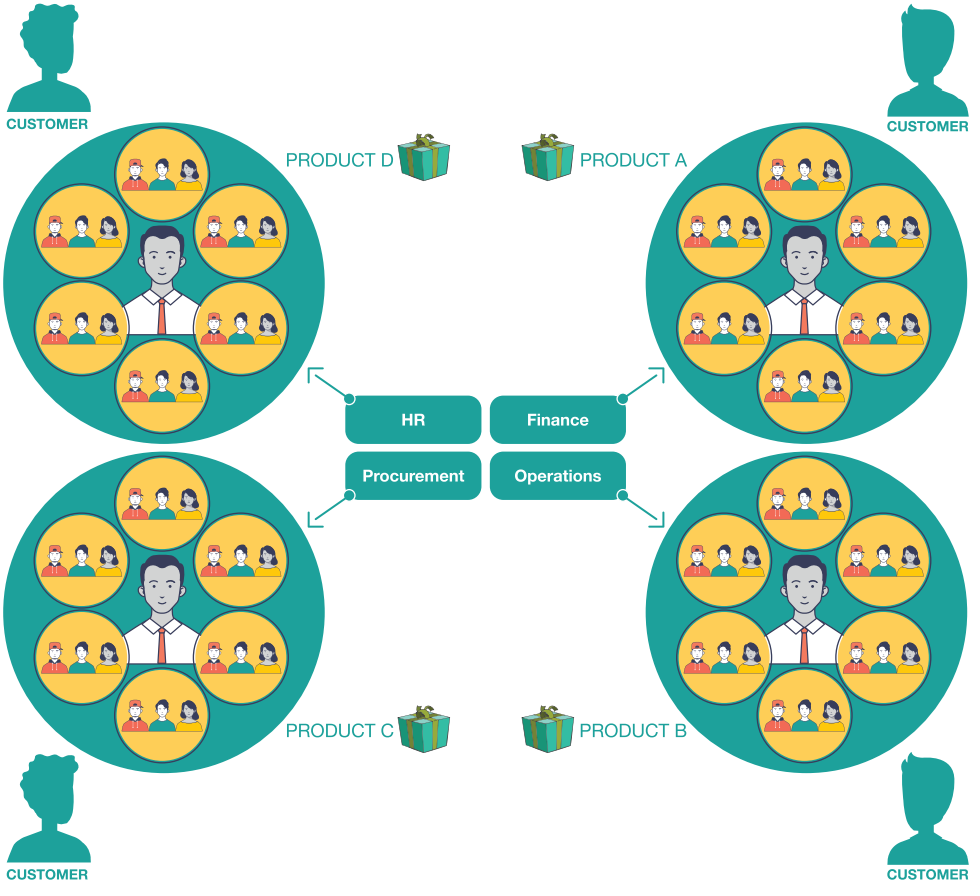

Figure 1.5
Prototype of an Agile organization.

When a product group consists mainly of feature teams, then coordination costs go down. Why? Teams that belong to the same product unit coordinate effectively because they report to the same manager, adopt the same goals and priorities, and share the same resources. The flow of work also increases because by grouping interdependent roles in the product units and in the teams, the teams waste less time on alignment and coordination activities. Over time, members of the product group are likely to develop a shared culture, further facilitating collaboration.

In general, you would expect to see the following structures and responsibilities in an Agile organization:

- Semi-independent product groups with separate leadership, finances, resources, and people, which may be augmented with shared services such as purchasing, sales, human operations, and finance.
- A person at the senior management level with a deep understanding of the product and process leading the product group.
- Operational management that is responsible for developing people through mentoring and improving the system of work.
- Teams that are responsible for delivering quality products. Working in such a team encourages shared team responsibility instead of individual responsibility for doing a specialized task. The teams are organized so that they have all the skills required to provide end-to-end value for customers without depending on people outside the team. Independence enables autonomous and fast-moving organizations.
- People who take responsibility for their process improvement. They own and solve the problems that occur in their daily work to improve agility.

Characteristics of Processes and People Practices

Which processes, people practices, and measures can you observe in an Agile organization? In Chapter 4, "Agile Organizational Design," we share an extensive overview. For now, Table 1.1 provides a brief summary.

Table 1.1 Typical Agile Organization Characteristics

Key Processes	People Practices	Measures
Teams are kept stable to allow them to develop into high-performing teams. It takes time for a team to grow and for people to develop a feeling of ownership of their process and product.	Operating in cross-functional teams means that the team members are willing and able to pick up work outside of their main expertise so as to always make progress as a team. Individuals have their deep expertise but also develop skills in different disciplines to become multifunctional specialists. Multifunctional specialists reduce bottlenecks created by the work imbalance across the different functions.	Team success is measured based on business outcome measures, not delivery outputs. The measures are derived from the purpose of the product or service and are expressed in terms of customer value and business impact.

Key Processes	People Practices	Measures
The teams work closely with the end users or (internal) customers to thoroughly understand their problems and needs. They use that learning to decide how to develop and improve their product or service.	People receive frequent feedback on their work. They use this feedback to develop their competence as well as their skills. Developing people is considered a key management focus.	The group is aware of their process effectiveness so that they can improve it. A more effective process will free up resources that can be either reinvested, relocated, or removed from the fixed cost.
Teams frequently validate the business and development assumptions made during development. The short feedback loops reduce risks and increase the possibilities to adapt effectively toward new insights.	People feel safe to fail, and there is room to learn. A two-year study by Google came to the following conclusion: "The highest-performing teams have one thing in common: psychological safety, the belief that you won't be punished when you make a mistake."[14]	
	Most people have a growth mindset and a passion for learning; they seek mastery of their abilities. The people value challenges and use them to put in the effort to learn and grow.[15]	

The organization's top management should clearly and unambiguously define what the optimization goals are and communicate those to members of the organization. Otherwise, the inconsistency between stated optimization goals and supporting organizational design decisions will create tensions and frustration in the field.

> *All teams will henceforth expose their data and functionality through services interfaces. Anyone who doesn't do this will be fired. Have a nice day.* —**Jeff Bezos, 2002 architecture mandate for Amazon.com**

Summary

Working with people who understand the *why* behind the new way of working is a crucial ingredient and lays the foundations for success. Management alignment on the optimization goals as well as co-creating an understanding of Agile principles and practices is an excellent first step to take.

The organizational design used severely impacts the organization's agility at the overall product level. The espoused optimizing goals must be supported by the organizational design to scale agility successfully. Unfortunately, when your current organizational design is based on a functional hierarchy, then you are likely optimizing for predictability, control, and resource utilization, and maintaining the status quo.

In contrast, agility at scale requires an organizational redesign so that you optimize for the right optimization goals. A senior manager who fills the role of organizational designer is needed.

References

1. KPMG. "Business Transformation and the Corporate Agenda." https://advisory.kpmg.us/articles/2017/business-transformation-and-the-corporate-agenda.html.
2. Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. *Best Practices in Product Innovation: What Distinguishes Top Performers* (Stage-Gate, 2003).
3. Peter Vosshall and Werner Vogels. "Dynamo: Amazon's Highly Available Key–Value Store." *ACM SIGOPS Operating Systems Review*, October 2007.
4. C. Koornhof. "Financial Flexibility and the Assessment of Future Cash Flows." *Investment Analyst Journal* 31 (1988): 13–19.
5. Peter Scholtes. *The Leader's Handbook: Making Things Happen, Getting Things Done* (McGraw-Hill Education, 1997), 2.
6. John Kotter. "Hierarchy and Network: Two Structures, One Organization." *Harvard Business Review.* https://hbr.org/2011/05/two-structures-one-organizatio.
7. Cesário Ramos. "Scale Your Product NOT Your Scrum." February 2016. https://www.agilix.nl/resources/ScaleYourProductNotYourScrum.pdf.
8. K. Schwaber. "SCRUM Development Process." In J. Sutherland, C, Casanave, J. Miller, P. Patel, and G. Hollowell, eds. *Business Object Design and Implementation* (Springer, 1997). https://doi.org/10.1007/978-1-4471-0947-1_11
9. "The 2020 Scrum Guide." https://scrumguides.org/scrum-guide.html.
10. Susan A. Wheelan. *Creating Effective Teams: A Guide for Members and Leaders*, 5th ed. (Sage, 2021), Chapter 1, p. 2.
11. Peter G. Northouse. *Leadership: Theory and Practice* (Sage Publications, 1997).
12. Jon R. Katzenbach. *The Wisdom of Teams: Creating the High-Performance Organization* (Harvard Business Review Press, October 13, 2015), 45.
13. Craig Larman and Bas Vodde. "Feature Team Primer." 2010. https://featureteams.org/feature_team_primer.pdf.
14. ReWork. "Guide: Understand Team Effectiveness." https://rework.withgoogle.com/guides/understanding-team-effectiveness/steps/introduction/.
15. Carol S. Dweck. *Mindset: The New Psychology of Success* (Ballantine Books, 2007).

Index

Symbols

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q

R

S

T

U

V

W-X-Y-Z