



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).

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors 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: 2022938926

Copyright © 2023 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

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

- Foreword by Craig Larman** **xix**
- Foreword by Dave West** **xxi**
- Preface** **xxiii**
- Introduction** **xxvii**
- Acknowledgments** **xxxv**
- About the Authors** **xxxvii**

- Part I Foundational Concepts** **1**
- 1 Organizing for Adaptability** **3**
- Hello, VUCA World! 3
- What Is an Agile Organization Design? 4
- What Does Your Organization Optimize For? 5
- Optimization Goals and Indirect Wishes 5
- Optimizing Goals of an Agile Organization 6
- Typical Problems When Adopting Agility 6
- The Functional Hierarchy Organization Design 7
- Avoid Copy–Paste Scaling: A Typical Scaling Approach 10
- Overview of an Agile Organization Design 11
- What Is an Agile Team? 12
- Structural Characteristics 13
- Summary 16
- References 17
- 2 Systems Thinking** **19**
- Basics of Systems Thinking 19
- Two Systems 19
- When to Apply Systems Thinking and Which Problem Does It Solve? 20
- Systems and Collections 21
- Organizations as Complex Social Systems 21
- Linear and Systems Thinking 23

Define the Boundary of the System Wisely	24
Work on the System	25
Iceberg of Systems Thinking	27
Analyzing the Current Situation	29
Organizational Learning and Creative Tension	30
Applying Systems Thinking	31
Causal Loop Diagram	31
A CLD Example	33
Shared Responsibility	34
Doom Loop Technique	35
Example: Using the Doom Loop Technique	36
Optimize the Whole	37
System Archetypes	38
Summary	48
References	49
3 Optimize for Adaptiveness	51
What It Means to Be Adaptable	53
Flow Efficiency	54
Behavior of Queues	54
Relationships Among Utilization, Cycle Time, and Variability	55
Queues Create Waste	57
Balancing Excess Capacity with the Cost of Delay	57
Relationships Among Utilization, Batch Size, and Cycle Time	58
One-Piece Continuous Flow	60
Queueing System Structures	61
Applying Queueing Structure to Agile Product Development	62
Algorithm for Dealing with Queues	62
Choose an Operational Strategy	63
Resource Efficiency	64
Flow Efficiency	65

Summary	67
References	68
4 Agile Organizational Design	69
Why Organizational Redesign?	70
Business Strategy Drives Organizational Design	71
Overview of a Design Process	72
Agile Organization Design Guidelines	73
Structural Guidelines	73
Process and Integration Guidelines	89
People Guidelines	97
Agile vs. Traditional Human Resources Management	101
Summary	104
References	105
5 An Agile Adoption Approach	107
Principles of an Agile Adoption	107
Overview of an Adoption Approach	109
A Bird's-Eye View of Adoption Activities	109
Overarching Adoption Guidelines	111
Form a Leading Group	111
Align Leadership on the Strategic Focus	112
Create Basic Conditions for Success	113
Learn Fast and Expand Change	115
Start with a Discontinuous Improvement Step	117
Avoid Providing the Answer Before You Know the Question	118
Applying Systems Thinking in Organizational Coaching	120
Agree on a Common System Optimization Goal	120
Address Secondary Concerns Without Compromising the System Goal	122
Have an Intermediate Improvement Goal and Perfection Vision	123
Planning Organizational Interventions	124

Summary	130
References	131
6 Coaching for Change	133
Guideline 1: Influence People	134
Use Corridor Politics	134
Show Personal Benefits	134
Build Rapport	135
Reach for Connectors	136
Guideline 2: Co-Create the Change	136
Guideline 3: Voluntary Participation	139
Get the Right People on the Bus and off the Bus	140
What It Means to Be a Volunteer	141
Nonviolent Education	143
What If You Don't Have Volunteers?	144
Guideline 4: Acknowledge the Loss	144
Beware of the Loss Aversion Bias	146
Guideline 5: Deliver the Message Continually	147
Guideline 6: Help People to Cross the Edge	149
Guest Visit	151
Create Edge Awareness	151
Honor the Past	153
Guideline 7: Axes of Change	153
All Components of the Strategy Are Important	154
The Four Axes of Change Model	154
Applying the Axes of Change to Teams	157
Guideline 8: Find the Right Balance of Radical/Incremental Change	159
Create a Change Roadmap	160
Dissolve the Problems Instead of Solving Them	161
Summary	161
References	162

Part II	Applying the Concepts	165
7	Group Facilitation	167
	Principles of Facilitation	167
	The Role of the Facilitator	168
	Meeting Goals	168
	Workshop Plan	168
	Ground Rules	169
	Consensus	169
	Parking Lot	170
	Track and Prioritize Ideas	170
	Dealing with Specific Behaviors	171
	Workshop Design	172
	The Plan Goes Awry	172
	The Magic of Engagement	173
	What Is a Serious Game?	174
	The Flow of the New Workshop	175
	The New Design Worked	177
	How to Design Your Workshop	177
	Facilitating Scrum Events	178
	Facilitating Sprint Planning	178
	Facilitating Product Backlog Refinement	184
	Facilitating Multi-Team Sprint Review	190
	Facilitating a Multi-Team Retrospective	192
	Summary	195
	References	196
8	Preparing the Product Group	197
	Areas of Concern for Successful Preparation	197
	Area 1: Involve the Managers	198
	Educating Management to Change Their Mental Models	199

Education Objectives	201
Getting Sufficient Managers' Support	202
Area 2: Understand the Current Reality	204
Go See	205
Individual Interviews	207
Value Stream Mapping	209
Scott Morton Diagram	212
Weighted SWOT and Confrontation Matrix	214
Finding System Structures Workshop	216
Aligning on the Go See Results with the Systems Model	218
Area 3: Create a Vision of the Future	221
Alignment Workshops	221
Shared Team Expectations Workshop	227
Shared Optimizing Goals Workshop	227
Perfection Vision Workshop	231
Progress So Far	233
Area 4: Identify the Product Group	233
Organize for Value Delivery	234
What About Internal Products?	236
Product Definition from Different Perspectives	237
Where Does the Product Definition Fit in Your Transformation?	237
How to Define the Product	238
Organizational Design to Eliminate Dependencies	242
Start with the Product Features	243
Using the Hotspots to Design Your Group	243
Summary	249
References	250
9 Launching the Product Group	251
Menu of Available Choices	252
Who Owns the Launching Process?	253
How Much Time Should It Take?	253

Initial Product Backlog Refinement	253
Try Features on the Product Backlog	253
How to Represent Features	254
Example of Creating an Initial Product Backlog with Product Goals and Sprint Goals	254
Define the Definition of Done	257
Component Team Velocity?	258
A Definition of Done Workshop	259
Feature Team Adoption Map	260
Choose Subsystem Names Close to the Customer Domain	261
Ordering the Components and DoD Activities	262
A Feature Team Adoption Map Workshop	262
Self-Designing Team Workshop	265
Consider a Market of Skills Workshop as Preparation	266
A Self-Designing Team Workshop	267
Team Lift-Off	270
The Lift-Off to Lay the Foundation for Success	271
Your Role as a Scrum Master	271
Discover a Shared Purpose	271
Align on Shared Values and Behavior	273
Facilitating Decision Making	277
Identify and Launch Communities	282
Identify Coordination Mechanisms	284
Create Useful Checklists	286
Sprint Goal Checklist	286
Underutilization Checklist	287
Emergent Work Checklist	288
Summary	288
References	289

10	Coaching Teams	291
	The Building Blocks of the Agile Organization	291
	Observations of an Agile Team	292
	What Makes an Agile Team?	292
	What You Can Do to Help a Group Become a Team	295
	Building on the Team Lift-Off: Coaching Multifunctional Learning	296
	Individual and Team Performance	297
	Multifunctional Learning and Agility	298
	Optimizing Value and Multifunctional Learning	302
	Culture of Narrow Specialists	304
	How to Optimize for Multifunctional Learning	305
	Coaching Multifunctional Learning Summary	315
	Dealing with Recurrent Complex Problems in Teams: Systems	
	Team Coaching	315
	Insights from Systems Coaching	317
	Perform Circular Observations to Identify Patterns of Interactions	317
	Team's System Assessment	322
	Improving Team Dynamics: Trust	329
	ORID: Focused Discussion	330
	Radical Candor	330
	Facilitating Conflict	331
	Facilitate Feedback Loops for Learning: Learning New Practices	332
	Step 1: Determine Your Goal	333
	Step 2: Create Specific Actions	334
	Step 3: Define Measures That Drive Learning	334
	Summary	335
	References	335
11	Guiding the Product Ownership	337
	What Makes a Team Product Owner?	337
	Avoid Product-Part Owners	338
	Replace Product-Part Owners with a Real Product Owner	339

How to Promote the Team Product Owner to Team Member	340
The Product Owner in a Senior Position	341
Avoid Product Owner Committees	342
Outward-Focused Product Owner	342
Workshop to Separate Product Owner Activities from General Managerial Activities	343
Product Ownership Measures	345
Measuring Business Impact	345
What About Budgets and Costs?	345
Measuring Team Performance	346
Measuring Product Qualities	347
Customer Understanding	347
Validate Assumptions Early and Often	348
The Product Owner Process	349
Serious Games for Customer Understanding	351
Stakeholder Alignment	351
Align on the Product Vision	351
Consider Co-Creating the Product Vision	352
Align on the Objectives	353
Product Owner Leadership	354
Try to Keep the Teams Stable	354
Reliable Forecasts	355
Avoid Micromanaging the Team	355
Collaborate as a Peer with the Teams	355
Stakeholder Committees	356
Connect the Teams with Users	356
Summary	356
References	357
Appendix Case Studies and Workshop Examples	359
Index	413

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

—Iliia 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 components¹:

- **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 Scrum⁴ 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>.

Register your copy of *Creating Agile Organizations* 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/register and log in or create an account. Enter the product ISBN (9780135853191) and 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.

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.

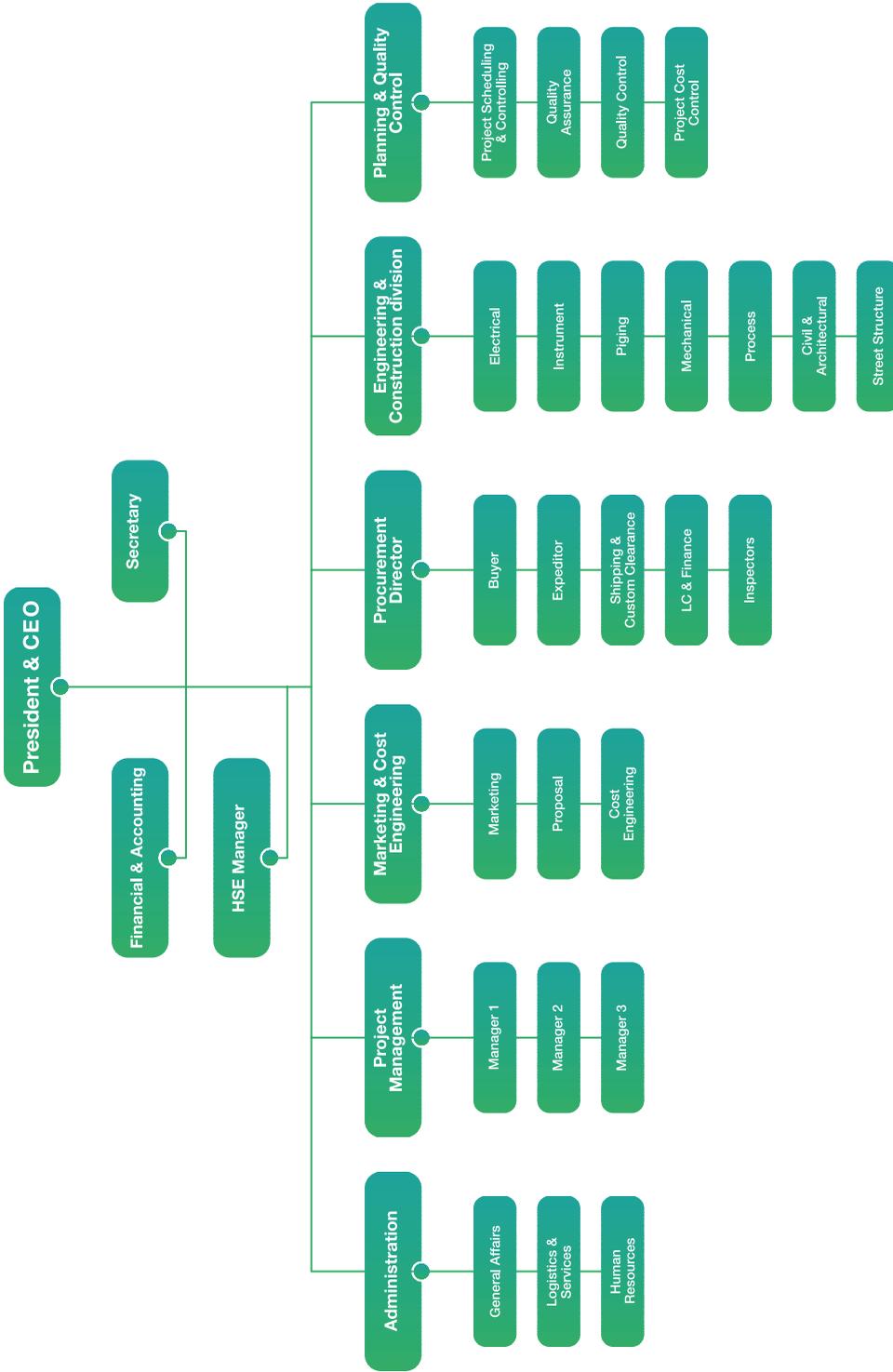


Figure 1.2
Hierarchy organization design.

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.

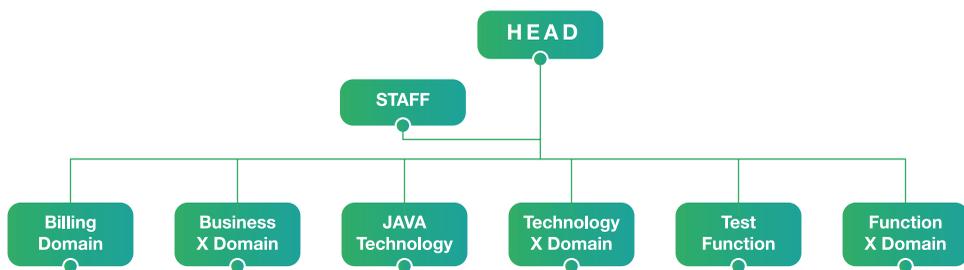


Figure 1.3
Typical silo organizational design in modern times.

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. Initially, 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).

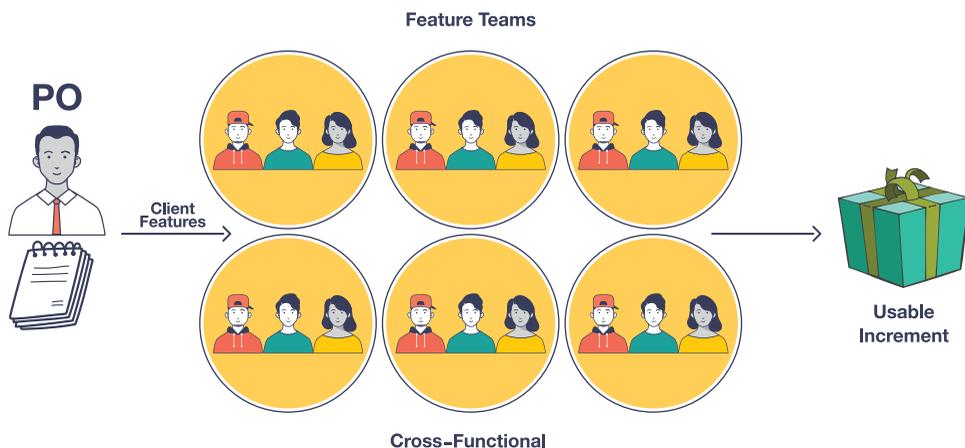


Figure 1.4
Feature teams.

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.

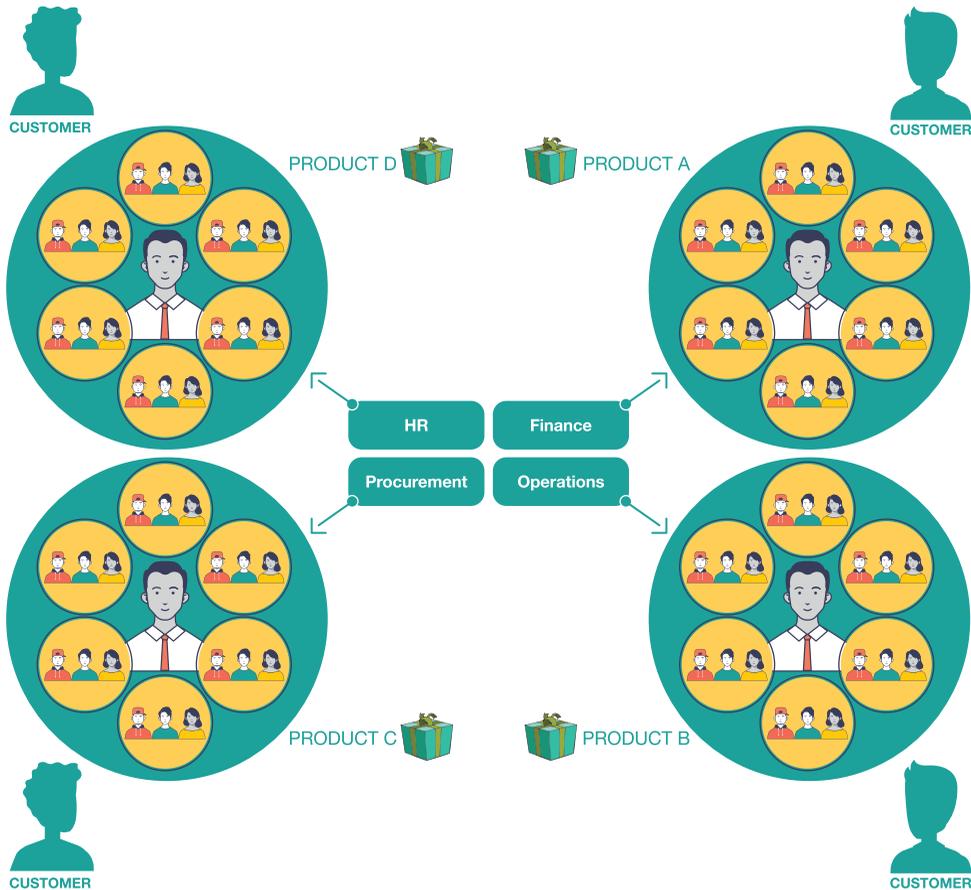


Figure 1.5
Prototype of an Agile organization.

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

%C&A (percent complete and accurate), 210
1-2-4-All structure, 214, 224, 228, 275, 392, 394, 397
The 5 Dysfunctions of a Team (Lencioni), 293
15% Solutions structure, 151, 176, 267, 325, 392
20/20 Vision, 277

A

ability, trust and, 391
accidental specialization, 249
accommodation, 397–399
accountability, 397
acknowledgement of loss, 144–147
 loss aversion bias, 146–147
 volume of loss, assessment of, 143
Ackoff, R. A. 79
Ackoff, Russell L. 21, 79, 80, 118, 315–316, 361
actions
 creating, 334
 feedback loops, 334
 improvement, 375–379
adaptability, 51–68
 definition of, 53–54
 flow efficiency, 54, 65–66
 importance of, 51–52
 operational strategy, 63
 as optimization goal, 5, 6, 62
 queueing theory
 algorithms, 54–62
 behavior of queues, 54–61
 queueing system structures, 61–62
 resource efficiency, 64–65
adoption, Agile. *See* Agile adoption
Adventures in Complexity (Kuhn), 117
aggression, obnoxious, 401
Agile adoption
 common mistakes in, 107
 conditions for success, 113–115
 change leaders, 114
 feedback loops, 115
 goals and objectives, 114–115
 safe spaces, 115
 expansion of
 continual improvement steps, 118
 discontinuous improvement steps, 117–118
 individual and organizational learning, 115–117
 leading groups, 111–112
 organizational interventions, 124–130
 BOTG (behavior over time graph), 29–30, 124, 126, 371
 CLD (Causal Loop Diagram), 31–35, 124, 126–130, 194, 218–219, 375–377
 experimentation, inspection, and adaption, 130
 high-leverage interventions, 130
 stories, creating, 125–126
 overview of, 109–111

- principles of, 107–109
- spreadsheet Agile adoption, 118–120
- standards in, 120
- strategic focus, 112–113
- systems thinking in, 120
 - immediate improvement goals, 122
 - perfection vision, 123–124
 - secondary concerns, 122
 - system optimization goals, 120–121
- Agile Manifesto*, 72, 142, 147, 190, 201
- Agile organizational design
 - agility at scale in, 13–15
 - business strategy as driver of, 71
 - cautions for, 71
 - common problems with, 6–7
 - copy-paste scaling, 10–11, 62, 204
 - definition of, 4–5
 - design process overview, 72–73
 - functional hierarchy compared to, 7–10
 - goals of, 70
 - importance of, 69
 - measures of success in, 15–16
 - optimization goals for
 - examples of, 5
 - indirect wishes versus, 5–6
 - selection of, 6
 - overview of, 11–12
 - people guidelines for, 97–101
 - multi-skill development, 99–101
 - separation of product management and line management, 97–99
 - performance appraisals, 101–104
 - in Agile setting, 102–104
 - in traditional setting, 101–102
 - process and integration guidelines, 89–96
 - commodity platforms, 93–94
 - cost of delay, 57–58, 96, 402–410
 - critical dependency, 95–96
 - emergent coordination, creating conditions for, 89–92
 - service units, 94–95
 - shared design services, 94–96
 - uncertain dependency, 95
 - processes and people practices in, 15–16
 - prototype of, 13–15
 - service-oriented architecture, 4
 - structural guidelines, 73–89
 - essential interdependencies, merging units with, 79–81
 - front-back model, 82–84
 - product groups, 73–75
 - reciprocal task interdependencies, 85–86
 - required capabilities, deriving from strategic focus, 87–89
 - shared customers and performance indicators, 87
 - unit functions, 76–79
 - teams
 - advantages of, 13
 - feature, 13
 - processes and people practices in, 15–16
 - structural characteristics of, 13
- Agile Retrospectives* (Derby and Larsen), 155, 192
- Agile teams. *See* teams
- agility, multifunctional learning and, 298–302
- agreements
 - explicit, 314–315
 - underutilization, 315
- alignment workshops
 - Perfection Vision, 231–233
 - overview of, 231–232
 - reasons for using, 232
 - running, 232–233
 - Shared Objectives, 222–227
 - examples of, 222–227
 - overview of, 222
 - running, 222–227
 - Shared Optimizing Goals, 227–231
 - closing, 231
 - developing diagram for, 229–231
 - structuring invitation for, 228
 - Shared Team Expectations, 227
- ambiguity, Agile approach to, 3
- amygdala, 395
- Anderson, Dianna, 291
- Anderson, Merrill, 291
- animate systems, 21
- Appreciative Interviews, 275, 392–394

archetypes, system, 38–48
 definition of, 38–39
 “drifting goals” 46–48
 archetype structure, 46–47
 leverage points and strategies, 48
 “fixes that backfire” 39–41
 archetype structure, 34, 39–40, 52
 leverage points and strategies, 41
 “limits to growth” 41–43
 archetype structure, 41–43
 leverage points and strategies, 43
 “shifting the burden” 43–46, 125, 128–129
 archetype structure, 43–45
 leverage points and strategies, 46
The Art of Focused Conversation (Stanfield), 389
 assumptions, validation of, 348
 authority, front-back model for, 82–84
 Autonomous Team pattern, 125
 avoiding conflict, 397–399
 Axes of Change model, 153–158
 application to teams, 157–158
 Creation phase, 156–157
 Decision phase, 156
 importance of components of, 150
 Motivation phase, 151
 overview of, 153–154
 Strengthen phase, 153–154

B

backlogs
 multiple implicit, 298–302
 Product. See Product Backlog
 Sprint Backlog, 62
 balancing loops, 33–34. *See also* CLD (Causal Loop Diagram)
 batch size, 54, 58–59
 Beedle, Mike, 3, 6
 Beer, Stafford, 79
 behavior
 behavior patterns, 397
 challenging behaviors, dealing with, 171
 of queues, 54–61
 cost of delay, 57–58, 96, 402–410

 one-piece continuous flows, 60–61
 terminology for, 54
 utilization, batch size, and cycle time, 58–59
 utilization, cycle time, and variability, 55–56
 waste, 57
 shared, 276–277
 behavior over time graph (BOTG), 29–30, 126, 371
 benevolence, 391
 Bens, Ingrid, 167
 Berlin, Isaiah, 28
 Bezos, Jeff, 16
 bias
 confirmation, 375–377
 loss aversion, 146–147
 Boria, Giacomo, 85
 BOTG (behavior over time graph), 29–30, 124, 126, 371
 bottlenecks
 alignment of resources to, 302–303
 individual and team performance and, 297–298
 boundaries, system, 24–25
 brainstorming, 397, 399
 Breuer, Christina, 391
 budgets, measures of, 345–346
 Buffett, Warren, 19
 business impact, measures of, 345
 business strategy, Agile organization design driven by, 71

C

candor, radical, 330–331, 400–402
 example of, 400–401
 manipulative insincerity, 401
 obnoxious aggression, 401
 practicing, 402
 ruinous empathy, 401
 capabilities, required, deriving from strategic focus, 87–89
 capacity, cost of delay and, 57–58, 402–410
 Cardon, Alain, 322
 cargo cults, 273
 Carse, James P. 139

- Causal Loop Diagram. *See* CLD (Causal Loop Diagram)
- cause-effect relationships, in CLD (Causal Loop Diagram), 32
- challenging behaviors, dealing with, 171
- change leaders, 114
- change management
 - acknowledgement of loss, 144–147
 - loss aversion bias, 146–147
 - volume of loss, assessment of, 143
- Axes of Change model, 153–158
 - application to teams, 157–158
 - Creation phase, 156–157
 - Decision phase, 156
 - importance of components of, 150
 - Motivation phase, 151
 - overview of, 153–154
 - Strengthen phase, 153–154
- co-creation of change, 136–138
- continuous message delivery, 144–145
- influencing of people, 134–136
 - connectors, 136
 - corridor politics, 134
 - personal benefits, 134
 - rapport, 135
- overview of, 133
- radical versus incremental change, 155–161
 - change roadmaps, 156–160
 - dissolution of problems in, 156–161
 - punctuated equilibrium theory, 155–156
- support for crossing the edge, 149
 - edge awareness, 151–152
 - edge model, 149–150
 - guest visits, 151
 - honoring the past in, 153
- voluntary participation, 139–141
 - characteristics of, 141
 - importance of, 139
 - lack of volunteers, 141
 - nonviolent education, 143
 - right/wrong people in, 140–141
- change roadmaps, 156–160
- change stories, creating, 225–227
- checklists, creation of, 252
 - emergent work checklist, 288
 - Sprint Goal checklist, 286
 - underutilization checklist, 287
- CLD (Causal Loop Diagram), 124, 194
 - benefits of, 35
 - CLD workshops, 218–219
 - confirmation bias and, 375–377
 - creating, 126–130
 - description of, 31–33
 - example of, 33–34
- coaching, 111
 - acknowledgement of loss, 144–147
 - loss aversion bias, 146–147
 - volume of loss, assessment of, 143
- Axes of Change model, 153–158
 - application to teams, 157–158
 - Creation phase, 156–157
 - Decision phase, 156
 - importance of components of, 150
 - Motivation phase, 151
 - overview of, 153–154
 - Strengthen phase, 153–154
- co-creation of change, 136–138
- conditions for team effectiveness, 291–292
- conflict, 398–400
 - Conflict Coaching sessions, 400
 - conflict-resolution guidelines, 399
 - questions to ask, 398–399
- continuous message delivery, 144–145
- individual and team performance, 297–298
- influencing of people, 134–136
 - connectors, 136
 - corridor politics, 134
 - personal benefits, 134
 - rapport, 135
- multifunctional learning, 305–315
 - agility and, 298–302
 - culture of narrow specialists, 304–305
 - enabling structures, 306–307
 - explicit agreements, 314–315
 - individual and team performance, 297–298
 - modes of development, 311–313
 - optimizing value strategy for, 302–303
 - overview of, 296

- questions and objections related to, 305–306
- slack time, 313–314
- star maps, 309–310
- summary of, 315
- visualization of work flow, 307–308
- overview of, 133, 296
- radical versus incremental change, 155–161
 - change roadmaps, 156–160
 - dissolution of problems in, 156–161
 - punctuated equilibrium theory, 155–156
- role of leadership in, 295–296
- social dynamics and, 292–295
 - commitment, 293
 - constructive conflict, 293
 - integrity and trust, 293–294
 - respect, 294
 - self-control, 294–295
- support for crossing the edge, 149
 - edge awareness, 151–152
 - edge model, 149–150
 - guest visits, 151
 - honoring the past in, 153
- systems team coaching, 315–329
 - description of, 315–317
 - example questions for, 321
 - insights from, 317
 - patterns of interactions in, 317–321
 - system assessments, 322–329
 - trust and, 329–335
 - undesired behavior, 322
- systems thinking in, 123–124
- voluntary participation, 139–141
 - characteristics of, 141
 - importance of, 139
 - lack of volunteers, 141
 - nonviolent education, 143
 - right/wrong people in, 140–141
- Coaching Change* (Hall), 153
- Cockburn, Alistair, 72
- co-creation of change, 136–138
- cognitive costs, 53
- COIN (context/observation/impact/next), 402
- collaboration, 355–356, 397–399
- Collins, Jim, 140
- co-location, creating, 92
- commitment, 275, 276, 277, 293
- committees
 - POs (Product Owners), 342
 - stakeholder, 356
- commodity platforms, 93–94
- communities
 - creating, 90–91
 - launching, 252, 282–284
- competing, 397–399
- complexity, Agile approach to, 3
- component specialized teams, 244–245
- component velocity, 258–259
- compromise, 397–399
- confirmation bias, 375–377
- conflict, constructive, 293
- Conflict Coaching sessions, 400
- conflict protocols, 397–398
- conflicts
 - conflict coaching, 398–400
 - Conflict Coaching sessions, 400
 - conflict-resolution guidelines, 399
 - questions to ask, 398–399
 - conflict protocols, 397–398
 - guidelines for resolving, 399
 - handling of, 331–332, 392–400
 - strategies for, 397–399
 - What Color Is the Conflict? exercise, 392
 - team toxins, 394–397
 - contempt, 394
 - criticism, 394
 - defensiveness, 395
 - stonewalling, 395
 - Team Toxins exercise, 395–397
- confrontation matrix, 214–215
- connectors, 136
- consensus, 169–170, 278–280, 398
 - benefits of, 278, 279–280
 - decision rules, 279–282
 - finding, 398
 - five-finger, 278–279
- constellations, trust, 392
- constraints, theory of, 297
- constructive conflict, 293

contempt, 394
 context, in Systems Assessment workshop, 324
 context/observation/impact/next (COIN), 402
 continual improvement steps, 118
 continuous flows, one-piece, 60–61
 continuous message delivery, 144–145
 Contract Game, 241–242
 Conversation Café, 143, 275, 392
 coordination, multi-team Sprint Planning, 183
 coordination mechanisms, 252, 284–286
 Coplien, James O.73
 copy-paste scaling, 10–11, 62, 204
 corridor politics, 134
 cost of delay, 57–58, 96, 402–410
 costs, measures of, 345–346
 courage, 275, 276, 277
 Cover Story, 385
 Covey, Stephen, 51
 COVID-19, 135, 151, 253
Creating Effective Teams (Wheelan), 11
 Creation phase, Axes of Change model, 156–157
 creative tension, 30–31
 critical dependency, 95–96
 criticism, 394
 cross-functional line managers, 98–99
 crossing the edge, support for, 149
 edge awareness, 151–152
 edge model, 149–150
 guest visits, 151
 honoring the past in, 153
 Csikszentmihalyi, Mihaly, 173
 CT (cycle time), 54, 118
 culture, definition of, 27
 culture of narrow specialists, 304–305
 current reality, understanding, 204–220
 CLDs (Causal Loop Diagrams). *See* CLD (Causal Loop Diagram)
 confrontation matrix, 214–215
 example of, 220
 Gemba, 205
 Go See, 205–207
 levels of understanding in, 205–206
 observation sessions, 206–207
 individual interviews, 207–209

keys for success, 205
 Scott Morton diagrams, 212–214
 overview of, 212–213
 Scott Morton workshops, 213–214
 system structures workshops, 216–218
 materials needed, 216
 structural invitation for, 216–218
 VSM (value stream mapping), 209–212
 metrics for, 210–211
 overview of, 209–210
 VSM workshop, 211–212
 WIP (work-in-progress), 211
 weighted SWOT analysis, 214–215
 customer understanding, 347–351
 assumptions, validation of, 348
 example of, 347
 games for, 351, 383–386
 stakeholder alignment
 on objectives, 353–354
 overview of, 351
 on product vision, 351–353
 truth curve, 349–350
 customer-centric organizations, 89
 customers, shared, 87
 cycle time (CT), 24, 54, 55–56, 118
 utilization and, 58–59
 variability and, 55–56

D

Daily Scrum, 125
 Darwin's theory of evolution, 156
 decision horizon, 21
 decision making, facilitation of, 252, 277–282
 consensus, 169–170, 278–280
 benefits of, 278, 279–280
 decision rules, 279–282
 finding, 398
 five-finger, 278–279
 decision matrix, 281
 Decision phase, Axes of Change model, 156
 decision rules, 279–282
 Decision-making phase (ORID method), 387–389
 decoupling of unit functions, 76–79
 defensiveness, 395

- Define Your Product workshop, 239–241
 - materials needed, 239
 - running, 239–241
 - structural invitation for, 239
- Definition of Done (DoD), 252, 257–262
 - component versus feature velocity, 258–259
 - concept of, 257–258
 - Definition of Done workshop, 259–260
 - participants, 259
 - running, 260
 - structuring invitation for, 260
- Definition of Done workshop, 259–260
 - participants, 259
 - running, 260
 - structuring invitation for, 260
- delay, cost of, 57–58, 96, 402–410
- delegation, 279
- deliberate practice, 332–335
- Deming, W. Edwards, 26
- dependencies, 26
 - critical, 95–96
 - organizational design to eliminate, 242
 - uncertain, 95
- Derby, Esther, 155, 192
- design, organization. *See* organization design
- design services, shared, 94–96
 - cost of delay, 57–58, 96, 402–410
 - critical dependency, 95–96
 - service units, 94–95
 - uncertain dependency, 95
- design structure matrix (DSM), 77
- deterministic cause-effect relationship, 32
- deterministic systems, 21
- development, modes of, 311–313
- developmental delay, cost of, 96, 402–410
- diagrams
 - BOTG (behavior over time graph), 124, 371
 - CLD (Causal Loop Diagram), 124, 194
 - benefits of, 35
 - CLD workshops, 218–219
 - description of, 31–33
 - example of, 33–34
 - confirmation bias and, 375–377
 - Scott Morton, 212–214
 - overview of, 212–213
 - Scott Morton workshops, 213–214
- disagreements
 - conflict coaching, 398–400
 - Conflict Coaching sessions, 400
 - conflict-resolution guidelines, 399
 - questions to ask, 398–399
 - conflict protocols, 397–398
 - guidelines for resolving, 399
 - handling of, 331–332, 392–400
 - strategies for, 397–399
 - What Color Is the Conflict? exercise, 392
 - team toxins, 394–397
 - contempt, 394
 - criticism, 394
 - defensiveness, 395
 - stonewalling, 395
 - Team Toxins exercise, 395–397
- discontinuous improvement steps, 117–118
- dissolving of problems, 156–161
- diverge-merge cycles, 183, 185, 260
- DoD (Definition of Done), 252, 257–262
 - component versus feature velocity, 258–259
 - concept of, 257–258
 - Definition of Done workshop, 259–260
 - participants, 259
 - running, 260
 - structuring invitation for, 260
- doom loop technique, 35–36
- Drawing Together, 392
- “drifting goals” archetype, 46–48
 - archetype structure, 46–47
 - leverage points and strategies, 48
- Drive* (Pink), 123
- DSM (design structure matrix), 77

E

- EAs (experiential aims), 389
- ecological systems, 22
- edge awareness, 151–152
- edge crossing, support for, 149
- edge awareness, 151–152

edge model, 149–150
 guest visits, 151
 honoring the past in, 153
 edge model, 149–150
 education
 nonviolent, 143
 of senior managers
 mental models, changing, 199–201
 objectives for, 201–202
 efficacy of teams, conditions for, 291–292
 efficiency, 54
 efficiency trap, 182
 flow, 54, 65–66
 resource, 54, 64–65
 Einstein, Albert, 369
 Ekman, Paul, 143
 Emergency Procedure pattern, 286
 emergent coordination, creating conditions for,
 89–92
 co-location, 92
 communities, 90–91
 overview of, 89–90
 temporary coordination roles, 91–92
 emergent work checklist, 288
 empathy, ruinous, 401
 enabling structures, for multifunctional learning,
 306–307
 engagement
 serious games for, 174–177, 383–386
 workshop, 172–177
 Ericsson, K. Anders, 334
 essential interdependencies, merging units with,
 79–81
 events
 definition of, 28
 discovering, 29–30
 evolution, theory of, 156
 experiential aims (EAs), 389
 experimentation, inspection, and adaption, 130
 expert leads, 99
 expertise, 99
 Experts, 171
 explicit agreements, 314–315

F

facilitation, group. *See* group facilitation
 facilitator, role of, 168
 Fahmy, Ahmad, 267
 Fake Scrum, 273
Fearless Change (Manns et al), 134
 feature owners, 10–11
 feature specialized teams, 244
 Feature Team Adoption Map (FTAM), 13, 252,
 257–258, 260–265
 axes, 260–261
 components and DoD activities, 262
 FTAM workshop, 262–265
 participants, 262
 running, 262–265
 structuring invitation for, 262
 overview of, 260–261
 subsystem names, 261–262
 feature velocity, 258–259
 features, in Product Backlog, 253–254
 feedback, importance of, 20–21
 Feedback Door technique, 215
 feedback loops, 112
 conditions for success, 115
 goals, 333
 overview of, 332–333
 FIFO (first in, first out), 64
The Fifth Discipline (Senge), 21, 135, 222
 fight or flight principle, 19–20, 395
 first in, first out (FIFO), 64
 Five Whys method, 375
 five-finger consensus, 278–279
 “fixes that backfire” archetype, 39–41
 archetype structure, 34, 39–40, 52
 leverage points and strategies, 41
 flexibility, as optimization goal, 6
 flow
 efficiency of, 54, 65–66
 one-piece continuous, 60–61
 visualization of, 307–308
 workshop, 172–177
Flow (Csikszentmihalyi), 173
 focus, 87–89, 112–113, 275, 276, 277
 Ford, Henry, 9

forecast reliability, 355
 Franklin, Melanie, 133
 front-back model, 82–84
 FTAM (Feature Team Adoption Map), 252, 257–258, 260–265
 axes, 260–261
 components and DoD activities, 262
 FTAM workshop, 262–265
 participants, 262
 running, 262–265
 structuring invitation for, 262
 overview of, 260–261
 subsystem names, 261–262
 full roulette, PBR (Product Backlog Refinement) with, 185
 functional hierarchy, 7–10
 overview of, 7
 separation of the head from the hands, 9
 silos organizational design, 9–10
 specialization of work, 9
 future, creating vision of, 221–233
 importance of, 221
 Perfection Vision workshop, 231–233
 overview of, 231–232
 reasons for using, 232
 running, 232–233
 Shared Objectives workshop, 222–227
 examples of, 222–227
 overview of, 222
 running, 222–227
 Shared Optimizing Goals workshop, 227–231
 closing, 231
 developing diagram for, 229–231
 structuring invitation for, 228
 Shared Team Expectations workshop, 227

G

Galbraith, Jay R.69, 70, 88
 Gallery Walk technique, 276
 games. *See also individual games*
 for customer understanding, 351, 383–386
 serious, 174–177, 383–386
Gamestorming (Gray, Brown, and Macanuso), 178
 Gather Data workshop, 369–370
 Gemba, 128–129, 205
 Giachetti, Ronald E.85
 Gilbert, Dan, 19–20
 Go See, 205–207
 example of, 220
 levels of understanding in, 205–206
 observation sessions, 206–207
 goals
 clarifying, 114–115
 determination of, 333
 immediate improvement goals, 123–124
 meeting, 168
 optimization
 examples of, 5
 indirect wishes versus, 5–6
 selection of, 6
 system optimization goals, 120–121
 Sprint Goals, 180, 286
 stakeholder alignment on, 353-CH11.0398
 system optimization, 120–121
Good to Great (Collins), 140
 Google Forms, 138
 Gorillas, 171
 Gottman, John, 394
 Graphic Game Plan game, 179–180
 Gray, Dave, 179
 Greatest Value:Create pattern, 114
 Griefwalking exercise, 143
 group facilitation
 principles of, 167–168
 role of facilitator in, 168
 Scrum events
 multi-team Retrospectives, 192–195
 multi-team Sprint Reviews, 189–192
 overview of, 178
 PBR (Product Backlog Refinement), 184–190
 Sprint Planning, 178–183
 techniques for
 challenging behaviors, dealing with, 171
 consensus, 169–170, 278–282
 ground rules, 169
 meeting goals, 168
 parking lot, 170

- prioritization and tracking of ideas, 170
- workshop plans, 168–169
- workshop design, 172–173
 - engagement and flow in, 172–177
 - serious games, 174–177, 383–386
 - structure of, 177–178
- group leadership, 111–112
- groups, product. *See* product groups
- groupthink, 277
- guest visits, 151
- guidelines, Agile organizational design
 - people, 97–101
 - multi-skill development, 99–101
 - separation of product management and line management, 97–99
 - process and integration, 89–96
 - commodity platforms, 93–94
 - emergent coordination, creating conditions for, 89–92
 - shared design services, 94–96
- structural, 73–89
 - essential interdependencies, merging units with, 79–81
 - front-back model, 82–84
 - product groups, 73–75
 - reciprocal task interdependencies, 85–86
 - required capabilities, deriving from strategic focus, 87–89
 - shared customers and performance indicators, 87
 - unit functions, 76–79

H

- Hackman, Richard J. 266, 291
- Hall, Michael, 153
- happiness door, 137
- hierarchy, functional, 7–10
 - overview of, 7
 - separation of the head from the hands, 9
 - silos organizational design, 9–10
 - specialization of work, 9
- high-leverage interventions, 130
- Hohman, Luke, 177

- honoring the past, 153
- human resources management
 - Agile versus traditional, 101–104
 - HR policies, 28
 - performance appraisals, 101–104
 - in Agile setting, 102–104
 - in traditional setting, 101–102

I

- iceberg of systems thinking, 27–29
- ideation, as optimization goal, 5
- Identify Patterns workshop, 371–372
- identifying product groups, 233–249
 - accidental specialization, 249
 - Define Your Product workshop, 239–241
 - materials needed, 239
 - running, 239–241
 - structural invitation for, 239
 - internal products, 236–237
 - organization for value delivery, 234–236
 - overview of, 233–234
 - product definition from different perspectives, 237
 - required organizational elements, 238–241
 - revenue stream, 238–241
 - role in transformation, 237
 - value area heat maps, 242–249
 - component specialized team, 244–245
 - feature specialized team, 244
 - optimal starting configuration, 245
 - organizational design to eliminate dependencies, 242
 - product features, 243
 - results of, 248–249
 - value areas in, 245–248
- immediate improvement goals, 123–124
- Impromptu Networking structure, 267, 284
- improvement action, formulating, 375–379
- improvement steps
 - continual, 118
 - discontinuous, 117–118
- incremental change, 155–161
 - change roadmaps, 156–160

- dissolution of problems in, 156–161
- punctuated equilibrium theory, 155–156
- indirect wishes, optimization goals versus, 5–6
- individual interviews, 207–209
- individual performance, coaching of, 297–298
- infinite games, 139
- influencing of people, 134–136
 - connectors, 136
 - corridor politics, 134
 - personal benefits, 134
 - rapport, 135
- Innovation Games* (Hohman), 177
- insincerity, manipulative, 401
- Institute of Cultural Affairs (ICA), 330
- integration guidelines, Agile organizational design, 87–96
 - commodity platforms, 93–94
 - emergent coordination, creating conditions for, 89–92
 - co-location, 92
 - communities, 90–91
 - overview of, 89–90
 - temporary coordination roles, 91–92
- shared design services, 94–96
 - cost of delay, 57–58, 96, 402–410
 - critical dependency, 95–96
 - service units, 94–95
 - uncertain dependency, 95
- integration test phase, 10–11
- integrity, 293–294, 391
- interactions, in systems team coaching
 - description of, 315–317
 - patterns of, 317–321
- interdependencies, 21
 - essential interdependencies, merging units with, 79–81
 - pooled, 85
 - reciprocal task, 85–86
 - sequential, 85
- internal products, product group identification, 236–237
- Interpretation phase (ORID method), 387–389
- interrelated parts, 21
- interviews

- Appreciative Interviews, 275, 392–394
- individual, 207–209
- investment uncertainty, 348

J-K

- Kahneman, Daniel, 19–20
- Kaikaku, 156
- Kanban, 157, 201, 258–259, 288, 380–381
- kata, 313
- Kates, Amy, 69
- Katzenbach, Jon R.12
- key performance indicators (KPIs), 27, 64
- Kilmann, Ralph, 393
- Kingman's formula, 55
- knowledge management, 108
- Kotter, John, 10
- KPIs (key performance indicators), 27, 64, 199
- Kramer, R. M.329
- Kuhn, Lesley, 117

L

- Large-Scale Scrum (LeSS), 51, 188, 200, 260, 284
- Larman, Craig, 51, 62, 91, 267
- Larsen, Diana, 155, 192, 251
- launching communities, 282–284
- launching product groups
 - checklists, creation of, 252
 - emergent work checklist, 288
 - Sprint Goal checklist, 286
 - underutilization checklist, 287
- coordination mechanisms, 252, 284–286
- decision making, 169–170, 252, 277–282, 398
- DoD (Definition of Done), 252, 257–262
 - component versus feature velocity, 258–259
 - concept of, 257–258
 - Definition of Done workshop, 259–260
- duration of, 253
- FTAM (Feature Team Adoption Map), 252, 257–258, 260–265
 - axes, 260–261
 - FTAM workshop, 262–265
 - overview of, 260–261
 - subsystem names, 261–262
- initial Product Backlog refinement, 253–257

- example of, 252, 254–257
 - features, 253–254
 - menu of available choices for, 252
 - MVL (Minimum Viable Launch), 252
 - ownership of launching process, 253
 - Self-Designing Team workshop, 265–270
 - Market of Skills workshop as preparation for, 266–267
 - overview of, 265–266
 - preparations, 267–269
 - running, 269–270
 - structuring invitation for, 269
 - team lift-off, 252, 270–277
 - as foundation for success, 271
 - overview of, 270–271
 - role of Scrum Master in, 271
 - shared purpose in, 271–273
 - Law of Two Feet, 283
 - lead time (LT), 210
 - The Leaders' Handbook* (Scholtes), 7
 - leadership
 - alignment on strategic focus, 112
 - change leaders, 114
 - coaching by. *See* coaching
 - POs (Product Owners), 354–356
 - collaboration with teams, 355–356
 - connection of teams with users, 356
 - forecast reliability, 355
 - micromanagement, avoidance of, 355
 - stakeholder committees, 356
 - team stability, 354
 - top-down, 108
 - leading groups, 111–112
 - Leading Teams* (Hackman), 266
 - Lean Coffee, 147, 284
 - lean thinking, 128–129
 - learning
 - costs of, 53
 - individual versus organizational, 115–117
 - measures as drivers of, 334–335
 - multifunctional, 305–315
 - agility and, 298–302
 - culture of narrow specialists, 304–305
 - enabling structures, 306–307
 - explicit agreements, 314–315
 - individual and team performance, 297–298
 - modes of development, 311–313
 - optimizing value strategy for, 302–303
 - overview of, 296
 - questions and objections related to, 305–306
 - slack time, 313–314
 - star maps, 309–310
 - summary of, 315
 - visualization of work flow, 307–308
 - as optimization goal, 5, 6
 - organizational, 30–31
 - Lencioni, Patrick, 293
 - LeSS (Large-Scale Scrum), 51, 188, 200, 260, 284
 - leverage points and strategies
 - “drifting goals” archetype, 48
 - “fixes that backfire” archetype, 41
 - “limits to growth” archetype, 43
 - “shifting the burden” archetype, 46
 - Lewin, Kurt, 197
 - lift-off workshops, 252
 - Liker, Jeffrey, 73
 - “limits to growth” archetype, 41–43
 - archetype structure, 41–43
 - leverage points and strategies, 43
 - line management, separation from product management, 97–99
 - linear thinking, 23
 - local optimizations, 37–38
 - Lombardi, Vince, 232
 - loops, feedback. *See* feedback loops
 - loss, acknowledgement of, 144–147
 - loss aversion bias, 146–147
 - volume of loss, assessment of, 143
 - loss aversion bias, 146–147
 - LT (lead time), 210
- ## M
-
- Mad Sad Glad game, 385
 - Mad Tea structure, 267
 - majority rule, 279
 - Making Questions Work* (Strachan), 322
 - managers, involvement in product group preparation, 198–204

- education objectives, 201–202
- mental models, changing, 199–201
- potential organizational impediments to, 198–199
- supportive versus unsupportive start situations, 202–204
- Managing the Design Factory* (Reinertsen), 57, 209, 303
- manipulative insincerity, 401
- maps
 - change roadmaps, 156–160
 - FTAM (Feature Team Adoption Map), 252, 257–258, 260–265
 - axes, 260–261
 - components and DoD activities, 262
 - FTAM workshop, 262–265
 - overview of, 260–261
 - subsystem names, 261–262
 - star, 309–310
 - value area heat maps, 242–249
 - component specialized team, 244–245
 - feature specialized team, 244
 - optimal starting configuration, 245
 - organizational design to eliminate, 242–249
 - product features, 243
 - results of, 248–249
 - value areas in, 245–248
 - VSMs (value stream maps), 65, 209–212
 - metrics for, 210–211
 - overview of, 209–210
 - VSM workshop, 211–212
 - WIP (work-in-progress), 211
- Market of Skills workshop, 266–267
- measure-for-improvement workshop, 387–390
 - materials needed, 382
 - Measures for Learning canvas, 383–386
 - structuring invitation for, 382
- measures, 15–16
 - as drivers of learning, 334–335
 - measure-for-improvement workshop, 387–390
 - product ownership, 345–347
 - budgets and costs, 345–346
 - business impact, 345
 - product qualities, 347
 - team performance, 346
 - team's measures for success, 273
- Measures for Learning canvas, 383–386
- Mechanical Scrum, 273
- meeting goals, 168
- mental models, 28, 32, 147
 - changing, 199–201
 - confirmation bias and, 375–377
 - discovering, 372–374
- Mentor role, 92
- message delivery, continuous, 144–145
- metrics, for VSM (value stream mapping), 210–211
- micromanagement, avoidance of, 355
- Min Specs structure, 277
- Minimum Viable Launch (MVL), 252
- Miro, 151
- mob programming, 311
- Modig, Niklas, 63
- Motivation phase, Axes of Change model, 150
- multifunctional learning, coaching of, 305–315
 - agility and, 298–302
 - culture of narrow specialists, 304–305
 - enabling structures, 306–307
 - explicit agreements, 314–315
 - individual and team performance, 297–298
 - modes of development, 311–313
 - optimizing value strategy for, 302–303
 - overview of, 296
 - questions and objections related to, 305–306
 - slack time, 313–314
 - star maps, 309–310
 - summary of, 315
 - visualization of work flow, 307–308
- multiple implicit backlogs, 298–302
- multi-skill development, 99–101
 - balance in, 101
 - creating conditions for, 100–101
 - proficiency and expertise, 99
- multi-team PBR (Product Backlog Refinement), 184–190
 - benefits of, 184
 - example of, 186–188
 - how to conduct, 184–185
 - initial setup for, 185–186

- role of users in, 185
- single-team versus, 188–190
- team representatives in, 187–190
- multi-team Retrospectives, facilitation of, 192–195
 - local versus system issues, 192
 - proactive learning setup, 191
 - regular mode, 192, 369
 - systemic mode, 194–195, 369–379
 - Gather Data workshop, 369–370
 - Identify Patterns workshop, 371–372
 - improvement action, formulating, 375–379
 - mental models, 372–374
- multi-team Sprint Planning, facilitation of, 180–183
 - closing summary, 183
 - coordination, 182–183
 - PBI selection, 182
 - recap, 180–182
- multi-team Sprint Reviews, facilitation of, 189–192
- multi-voting, 275, 279
- MVL (Minimum Viable Launch), 252
- The Mythical Man Month* (Brooks), 30

N

- narrow specialists, culture of, 304–305
- neocortex, 19–20
- nested systems, 22
- Nies, Ainsley, 251
- Nine Whys structure, 275
- nonviolent communication (NVC), 143, 402
- Northouse Peter G.12
- Now, How, Wow game, 176, 325
- number of people, 29
- Nuñez, Alba N.85
- NVC (nonviolent communication), 143, 402

O

- obedience stories, creating, 125–126
- Objective phase (ORID method), 387–389
- objectives, stakeholder alignment on, 353–354
- objectives and key results (OKRs), 341
- obnoxious aggression, 401
- observation sessions, 206–207
- Ohno, Taiichi, 303

- one-piece continuous flows, 60–61
- Onojighofia, Tobore, 147
- Open Space, 283–284
- openness, 275, 276, 277
- operational strategy, for adaptability, 63
- operations-centric organization, 89
- optimization
 - for adaptability, 51–68
 - definition of, 53–54
 - flow efficiency, 54, 65–66
 - importance of, 51–52
 - operational strategy, 63
 - as optimization goal, 5, 6
 - queueing theory, 54–62
 - resource efficiency, 64–65
 - goals
 - examples of, 5
 - indirect wishes versus, 5–6
 - selection of, 6
 - system optimization goals, 120–121
 - local, 37–38
 - for multifunctional learning, 305–315
 - enabling structures, 306–307
 - explicit agreements, 314–315
 - modes of development, 311–313
 - questions and objections related to, 305–306
 - slack time, 313–314
 - star maps, 309–310
 - visualization of work flow, 307–308
 - whole, 37–38
- optimizing value, multifunctional learning and, 302–303
- organization design, 69, 124–130. *See also* teams
 - Agile. *See* Agile organizational design
 - agility at scale in, 13–15
 - design process overview, 72–73
 - functional hierarchy, 7–10
 - overview of, 7
 - separation of the head from the hands, 9
 - silos organizational design, 9–10
 - specialization of work, 9
 - measures of success in, 15–16
 - processes and people practices in, 15–16
 - prototype of, 13–15
 - service-oriented architecture, 4

organization stress, 26–27

organizational coaching, systems thinking in, 120

- immediate improvement goals, 122
- perfection vision, 123–124
- secondary concerns, 122
- system optimization goals, 120–121

organizational elements, identifying, 238–241

organizational interventions, 124–130

- BOTG (behavior over time graph), 126
- CLD (Causal Loop Diagram), 126–130, 375–377
- experimentation, inspection, and adaption, 130
- high-leverage interventions, 130
- stories, creating, 125–126

organizational learning, 30–31

Organizations in Action (Thompson), 85

ORID method, 330, 386–391

- checklist for, 392–393
- framework questions, 387–389
- phases of, 386–387
- rational and experiential aims, 389
- summary of, 391

Out of the Crisis (Deming), 26

outward-focused POs (Product Owners), 342–344

Overall Refinement, 188

ownership

- of launching process, 253
- product, 111

P

pair programming, 311

parking lot, 170

partial roulette, PBR (Product Backlog Refinement) with, 185

participants

- Definition of Done workshop, 259
- FTAM (Feature Team Adoption Map) workshop, 262

past, honoring of, 153

Pattern Sequence workshop, 365–369

- application of patterns, 368–369
- creation of pattern sequence, 366–368
- materials needed, 365
- online resources for, 369
- selection of patterns to try, 365–366
- structural invitation for, 365

patterns, 27

- Autonomous Team, 125
- building team with, 361–362
- components of, 360
- definition of, 28, 360
- discovering, 29–30
- Emergency Procedure, 286
- expanding system boundaries with, 362–363
- pattern sequences, 363–369
 - pattern relationships, handling, 363–364
 - Pattern Sequence workshop, 365–369
 - starting context, finding, 364–365
- Scrum (Master) Coach, 128–129
- systemic approach to, 361
- Teams That Finish Early Accelerate Faster, 314
- Testable Improvements, 310
- when to use, 359

PBIs (Product Backlog Items), splitting, 381–383

- benefits of, 381–382
- importance of, 378–383
- principles of, 382–383
- results of, 383
- story splitting workshops, 378–383

PBR (Product Backlog Refinement), facilitation of, 184–190

- benefits of, 184
- example of, 186–188
- how to conduct, 184–185
- initial refinement of, 252, 253–257
- initial setup for, 185–186
- role of users in, 185
- single-team versus multi-team, 188–190
- team representatives in, 187–190

people, influencing, 134–136

- connectors, 136
- corridor politics, 134
- number of people, 29
- personal benefits, 134
- rapport, 135

people guidelines, Agile organizational design, 97–101

- characteristics of people practices, 15–16
- multi-skill development, 99–101

- balance in, 101
- creating conditions for, 100–101
- proficiency and expertise, 99
- separation of product management and line management, 97–99
- percent complete and accurate (%C&A), 210
- perfection game, 137
- Perfection Vision workshop, 123–124, 231–233
 - overview of, 231–232
 - reasons for using, 232
 - running, 232–233
- performance
 - individual, coaching of, 297–298
 - performance appraisals, 101–104
 - in Agile setting, 102–104
 - in traditional setting, 101–102
 - performance indicators, shared, 87
 - team, coaching of, 297–298
- permission, asking for, 399
- personal benefits, 134
- perturbations, 156
- Pikulev, Alexey, 391
- Pink Daniel, 123
- plans, workshop, 168
- pomodoro technique, 313
- pooled interdependencies, 85
- POs (Product Owners), 83
 - committees, 342
 - customer understanding, 347–351
 - assumptions, validation of, 348
 - example of, 347
 - games for, 351, 383–386
 - stakeholder alignment, 351–354
 - truth curve, 349–350
 - definition of, 337
 - inward- versus outward-focused, 342–344
 - leadership by, 354–356
 - collaboration with teams, 355–356
 - connection of teams with users, 356
 - forecast reliability, 355
 - micromanagement, avoidance of, 355
 - stakeholder committees, 356
 - team stability, 354
 - measures, 345–347
 - budgets and costs, 345–346
 - business impact, 345
 - measure-for-improvement workshop, 387–390
 - product qualities, 347
 - team performance, 346
- Product-Part Owners, 338–339
- promoting to team members, 340
- role in PBR (Product Backlog Refinement), 189
- in senior positions, 341
- workshop for, 343–344
- practice, deliberative, 334
- predictability, 391
- preparations, for Self-Designing Team workshop, 267–269
- preparing product groups, 233–249
 - current reality, understanding, 204–220
 - CLDs (Causal Loop Diagrams). *See* CLD (Causal Loop Diagram)
 - confrontation matrix, 214–215
 - example of, 220
 - Gemba, 205
 - Go See, 205–207
 - individual interviews, 207–209
 - keys for success, 205
 - Scott Morton diagrams, 212–214
 - system structures workshops, 216–218
 - VSM (value stream mapping), 209–212
 - weighted SWOT analysis, 214–215
 - current reality, understanding of, 204–220
 - manager involvement in, 198–204
 - education objectives, 201–202
 - mental models, changing, 199–201
 - potential organizational impediments to, 198–199
 - supportive versus unsupportive start situations, 202–204
- organization for value delivery, 234–236
- overview of, 197–198, 233–234
- product group identification, 233–249
 - accidental specialization, 249
 - Define Your Product workshop, 239–241
 - internal products, 236–237
 - overview of, 233–234

- product definition from different perspectives, 237
- required organizational elements, 238–241
- revenue stream, 241–242
- role in transformation, 237
- value area heat maps, 242–249
- vision of future, creation of, 221–233
 - importance of, 221
 - Perfection Vision workshop, 231–233
 - Shared Objectives workshop, 222–227
 - Shared Optimizing Goals workshop, 227–231
 - Shared Team Expectations workshop, 227
- The Principles of Product Development Flow* (Reinertsen), 26–27, 57, 60, 209
- prioritization of ideas, 170
- process and integration guidelines, Agile
 - organizational design, 89–96
 - commodity platforms, 93–94
 - emergent coordination, creating conditions for, 89–92
 - co-location, 92
 - communities, 90–91
 - overview of, 89–90
 - temporary coordination roles, 91–92
 - required capabilities, deriving from strategic focus, 89–92
 - shared design services, 94–96
 - cost of delay, 57–58, 96, 402–410
 - critical dependency, 95–96
 - service units, 94–95
 - uncertain dependency, 95
- process time (PT), 210
- processes, characteristics of, 15–16
- Product Backlog
 - multiple implicit backlogs, 298–302
 - PBR (Product Backlog Refinement), 184–190
 - benefits of, 184
 - example of, 186–188
 - how to conduct, 184–185
 - initial refinement of, 252, 253–257
 - initial setup for, 185–186
 - role of users in, 185
 - single-team versus multi-team, 188–190
 - team representatives in, 187–190
 - Product Backlog Items (PBIs), splitting, 381–383
 - benefits of, 381–382
 - importance of, 378–383
 - principles of, 382–383
 - results of, 383
 - story splitting workshops, 378–383
 - Product Backlog Items (PBIs), splitting, 381–383
 - benefits of, 381–382
 - importance of, 378–383
 - principles of, 382–383
 - results of, 383
 - story splitting workshops, 378–383
 - product definition
 - accidental specialization, 249
 - Define Your Product workshop, 239–241
 - materials needed, 239
 - running, 239–241
 - structural invitation for, 239
 - from different perspectives, 237
 - required organizational elements, 238–241
 - revenue stream, 241–242
 - role in transformation, 237
 - value area heat maps, 242–249
 - component specialized team, 244–245
 - feature specialized team, 244
 - optimal starting configuration, 245
 - organizational design to eliminate dependencies, 242
 - product features, 243
 - results of, 248–249
 - value areas in, 245–248
 - product features, in value area heat maps, 243
 - product group leads, 98
 - product group preparation, 233–249
 - current reality, understanding, 204–220
 - CLDs (Causal Loop Diagrams). *See* CLD (Causal Loop Diagram)
 - confrontation matrix, 214–215
 - example of, 220
 - Gemba, 205
 - Go See, 205–207
 - individual interviews, 207–209
 - keys for success, 205
 - Scott Morton diagrams, 212–214

- system structures workshops, 216–218
- VSM (value stream mapping), 209–212
- weighted SWOT analysis, 214–215
- current reality, understanding of, 204–220
- manager involvement in, 198–204
 - education objectives, 201–202
 - mental models, changing, 199–201
 - potential organizational impediments to, 198–199
 - supportive versus unsupportive start situations, 202–204
- overview of, 197–198
- product group identification, 233–249
 - accidental specialization, 249
 - Define Your Product workshop, 239–241
 - internal products, 236–237
 - organization for value delivery, 234–236
 - overview of, 233–234
 - product definition from different perspectives, 237
 - required organizational elements, 238–241
 - revenue stream, 238–241
 - role in transformation, 237
 - value area heat maps, 242–249
- vision of future, creation of, 221–233
 - importance of, 221
 - Perfection Vision workshop, 231–233
 - Shared Objectives workshop, 222–227
 - Shared Optimizing Goals workshop, 227–231
 - Shared Team Expectations workshop, 227
- product groups, 73–75
 - collaboration with, 117
 - identifying, 233–249
 - accidental specialization, 249
 - Define Your Product workshop, 239–241
 - internal products, 236–237
 - organization for value delivery, 234–236
 - overview of, 233–234
 - product definition from different perspectives, 237
 - required organizational elements, 238–241
 - revenue stream, 241–242
 - role in transformation, 237
 - value area heat maps, 242–249
- launching
 - checklists, creation of, 252–288
 - communities, 282–284
 - coordination mechanisms, 252, 284–286
 - decision making, 252, 277–282
 - DoD (Definition of Done), 252, 257–262
 - duration of, 253
 - FTAM (Feature Team Adoption Map), 252, 260–265
 - initial Product Backlog refinement, 253–257
 - menu of available choices for, 252
 - MVL (Minimum Viable Launch), 252
 - ownership of launching process, 253
 - Self-Designing Team workshop, 265–270
 - team lift-off, 252, 270–277
- number of, 116
- preparing, 109–111, 233–249. *See also* product group preparation
 - current reality, understanding of, 204–220
 - manager involvement in, 198–204
 - overview of, 197–198
 - vision of future, creation of, 221–233
- product management, separation from line management, 97–99
- Product Organization, 361
- Product Owners. *See* POs (Product Owners)
- product ownership, 111
- product qualities, measures of, 347
- product vision, stakeholder alignment on, 351–353
- product-centric organizations, 88
- Product-Part Owners, 338–339
- proficiency, 99
- project managers, 10–11
- protocols, conflict, 397–398
- Pseudo-Scrum, 273
- psychological safety, 273–274
- PT (process time), 210
- punctuated equilibrium theory, 155–156
- pure waste, 53
- purpose, shared, 271–273
 - alignment on shared values/behavior, 273–277
 - clarification of values, 276
 - identification of behavior patterns, 276–277
 - identification of values, 275
 - psychological safety and trust, 273–274

questions to ask, 271–273
 team's actions and expected outputs, 272
 team's measures for success, 273

Q

quality, as optimization goal, 5
 queueing, algorithms for, 62
 queueing theory
 algorithms, 62
 behavior of queues, 54–61
 cost of delay, 57–58, 96, 402–410
 flow efficiency, 54
 one-piece continuous flows, 60–61
 terminology for, 54
 utilization, batch size, and cycle time, 58–59
 utilization, cycle time, and variability, 55–56
 waste, 57
 queueing system structures, 61–62

R

radical candor, 330–331, 400–402
 example of, 400–401
 manipulative insincerity, 401
 obnoxious aggression, 401
 practicing, 402
 ruinous empathy, 401
 radical change, 155–161
 change roadmaps, 156–160
 dissolution of problems in, 156–161
 punctuated equilibrium theory, 155–156
 Ramos, Cesário, 10, 62
 rapport, building, 136
 RAs (rational aims), 389
 reality, understanding. *See* current reality, understanding
 reciprocal task interdependencies, 85–86
Re-creating the Corporation (Ackoff), 21, 79, 118
 refinement of Product Backlog, 253–257
 example of, 252, 254–257
 features, 253–254
 Reflection canvas, in Systems Assessment workshop, 324–325
 Reflexive phase (ORID method), 387–389

Reinertsen, Donald G. 26–27, 57, 60, 209, 303
 reinforcing loops, CLD (Causal Loop Diagram), 32
 relationships
 pattern, 363–364
 in systems team coaching, 315–317
 release management, 10–11
 release managers, 25
 release quality, 29
 reliability
 of forecasts, 355
 as optimization goal, 5
 required capabilities, deriving from strategic focus, 87–89
 required organizational elements, identifying, 238–241
 resolving conflict, 399
 resource efficiency, 5, 54, 64–65
 respect, 275, 276, 277, 294
 responsibility
 front-back model for, 82–84
 responsibility process model, 30
 shared, 34–35
 Retrospectives, facilitation of, 192–195
 local versus system issues, 192
 proactive learning setup, 191
 regular mode, 192, 369
 systemic mode, 194–195, 369–379. *See also* systemic mode Sprint Retrospectives
 Return on Time Invested exercise, 176, 177, 215, 385
 revenue stream, clarification of, 238–241
 roles, temporary coordination, 91–92
 roulette, PBR (Product Backlog Refinement) with, 185
 roulette technique, 36
 ruinous empathy, 401
 rules
 decision, 279–282
 for group facilitation, 169

S

safe spaces, 115
 safety, as optimization goal, 5
 SBIs (Sprint Backlog Items), 62
 scaling

- agility at scale, 13–15
- copy-paste, 10–11, 62, 204
- scaling sequence, 386–387
- Scaling Lean and Agile Development* (Larman), 62
- schedule uncertainty, 348
- Scholtes, Peter, 7
- Schults, Kirsten, 150
- Schwaber, Ken, 31, 107, 257, 337
- scientific management, 26
- Scott Morton diagrams, 212–214
 - overview of, 212–213
 - Scott Morton workshops, 213–214
- Scrum (Master) Coach pattern, 128–129
- A Scrum Book* (Scrum Patterns Group), 92, 114, 242, 334, 361
- Scrum events, facilitation of
 - multi-team Retrospectives, 192–195
 - local versus system issues, 192
 - proactive learning setup, 191
 - regular mode, 192, 369
 - systemic mode, 194–195, 369–379. *See also* systemic mode Sprint Retrospectives
 - multi-team Sprint Reviews, 189–192
 - overview of, 178
 - PBR (Product Backlog Refinement), 184–190
 - benefits of, 184
 - example of, 186–188
 - how to conduct, 184–185
 - initial setup for, 185–186
 - role of users in, 185
 - single-team versus multi-team, 188–190
 - team representatives in, 187–190
 - Sprint Planning, 178–183
 - multi-team, 180–183
 - single-team, 179–180
- The Scrum Guide*, 83
- Scrum Masters, 28
 - dependence on, 126–128
 - role of, 125–126, 271
- Scrum patterns. *See* patterns
- scrumbook.org, 369
- Scrum-Buts, 31
- secondary concerns, 122
- security, as optimization goal, 5
- Seddon, John, 203
- self-control, 294–295
- Self-Designing Team workshop, 252, 265–270
 - Market of Skills workshop as preparation for, 266–267
 - overview of, 265–266
 - preparations, 267–269
 - running, 269–270
 - structuring invitation for, 269
- Senge, Peter M., 21, 135, 222
- separation of the head from the hands, 9
- sequence, scaling, 386–387
- sequences, pattern, 363–369
 - pattern relationships, handling, 363–364
 - Pattern Sequence workshop, 365–369
 - application of patterns, 368–369
 - creation of pattern sequence, 366–368
 - materials needed, 365
 - online resources for, 369
 - selection of patterns to try, 365–366
 - structural invitation for, 365
 - starting context, finding, 364–365
- sequential interdependencies, 85
- serious games, 174–177, 383–386
- servers, 54
- service time (ST), 54
- service units, 94–95
- service-oriented architecture, 4
- The Seven Principles for Making Marriage Work* (Gottman), 394
- shared behaviors, identification of, 276–277
- shared customers, 87
- shared design services, 94–96
 - cost of delay, 57–58, 96, 402–410
 - critical dependency, 95–96
 - service units, 94–95
 - uncertain dependency, 95
- Shared Objectives workshop, 222–227
 - examples of, 222–227
 - overview of, 222
 - running, 222–227
- Shared Optimizing Goals workshop, 227–231
 - closing, 231
 - developing diagram for, 229–231

- structuring invitation for, 228
- shared purpose, discovery of, 271–273
 - alignment on shared values/behavior, 273–277
 - clarification of values, 276
 - identification of behavior patterns, 276–277
 - identification of values, 275
 - psychological safety and trust, 273–274
 - questions to ask, 271–273
 - team's actions and expected outputs, 272
 - team's measures for success, 273
- shared responsibility, 34–35
- Shared Team Expectations workshop, 227
- shared values
 - alignment on, 273–277
 - clarification of, 276
 - identification of, 275
 - list of core human values, 275
- Shift & Share, 36, 260, 275, 277, 392
- “shifting the burden” archetype, 43–46, 125
 - archetype structure, 43–45, 125, 128–129
 - leverage points and strategies, 46
- short-term cause-effect relationship, 32
- silent brainstorming, 397
- silo organizational design, 9–10
- single-team Sprint Planning, facilitation of, 179–180
- situation analysis, 29–30
- situation/behavior/impact, 402
- slack time, 313–314
- small and medium entrepreneurs (SME), 74
- Snack Shrine pattern, 92
- social dynamics of teams, 292–295
 - commitment, 293
 - constructive conflict, 293
 - integrity and trust, 293–294
 - respect, 294
 - self-control, 294–295
- social systems, 22
- solution uncertainty, 348
- specialists, culture of narrow specialists, 304–305
- specialization of work, 9
- speed, as optimization goal, 5
- Speedboat game, 174, 176, 385
- Spiral Journal technique, 143
- splitting PBIs (Product Backlog Items), 381–383
 - benefits of, 381–382
 - importance of, 378–383
 - principles of, 382–383
 - results of, 383
 - story splitting workshops, 378–383
- Spotify model, 201
- spreadsheet Agile adoption, 118–120
- Sprint Backlog, 62
- Sprint Backlog Items (SBIs), 62
- Sprint Goals, 180, 286
- Sprint Planning, facilitation of, 178–183
 - multi-team, 180–183
 - closing summary, 183
 - coordination, 182–183
 - PBI selection, 182
 - recap, 180–182
 - single-team, 179–180
- Sprint Reviews, multi-team, 189–192
- Sprints, definition of, 11
- ST (service time), 54
- stability, team, 354
- stakeholder alignment
 - on objectives, 353–354
 - overview of, 351
 - on product vision, 351–353
 - stakeholder committees, 356
- standards, Agile adoption, 120
- Stanfield, Brian, 387–389
- star maps, 309–310
- starting context, finding, 364–365
- Stoelhorst, J. W.⁸⁷
- stonewalling, 395
- stories. *See* PBIs (Product Backlog Items), splitting
- story splitting workshops, 378–383
- Strachan, Dorothy, 322
- strategic focus, 87–89, 112–113
- Strengthen phase, Axes of Change model, 153–154
- stress, organizational, 26–27
- Strikwerda, J.⁸⁷
- structural guidelines, Agile organizational design, 73–89
 - essential interdependencies, merging units with, 79–81
 - front-back model, 82–84

- product groups, 73–75
- reciprocal task interdependencies, 85–86
- required capabilities, deriving from strategic focus, 87–89
- shared customers and performance indicators, 87
- unit functions, 76–79
- structures
 - definition of, 28
 - discovering, 29–30
- Stumbling on Happiness* (Gilbert), 19–20
- subsystem names, FTAM (Feature Team Adoption Map), 261–262
 - participants, 262
 - running, 262–265
 - structuring invitation for, 262
- success of Agile adoption, conditions for, 113–115
 - change leaders, 114
 - feedback loops, 115
 - goals and objectives, 114–115
 - measures of, 273
 - safe spaces, 115
- super-majority rule, 279
- support, managerial, 202–204
- The Surprising Power of Liberating Structures* (Lipmanowicz and McCandless), 178
- swarming, 311
- switching costs, 53
- SWOT analysis, 214–215
- system archetypes, 38–48
 - definition of, 38–39
 - “drifting goals” 46–48
 - archetype structure, 46–47
 - leverage points and strategies, 46–47
 - “fixes that backfire” 39–41
 - archetype structure, 39–40, 52
 - leverage points and strategies, 41
 - system optimization goals, 34
 - “limits to growth” 41–43
 - archetype structure, 41–43
 - leverage points and strategies, 43
 - “shifting the burden” 43–46, 125, 128–129
 - archetype structure, 43–45
 - leverage points and strategies, 46
- system assessments
 - objectives of, 322
 - Systems Assessment workshop, 323–329
- system boundaries, expanding, 362–363
- system specifications, 10–11
- system structures
 - queueing, 61–62
 - system structures workshops, 216–218
 - materials needed, 216
 - structural invitation for, 216–218
- systemic mode Sprint Retrospectives, 194–195, 369–379
 - Gather Data workshop, 369–370
 - Identify Patterns workshop, 371–372
 - improvement action, formulating, 375–379
- mental models
 - confirmation bias and, 375–377
 - discovering, 372–374
- Systemic Team and Organizational Coaching* (Cardon), 322
- systems
 - animate, 21
 - archetypes, 38–48
 - definition of, 38–39
 - “drifting goals” 46–48
 - “fixes that backfire” 39–41
 - “limits to growth” 41–43
 - boundaries, 24–25
 - definition of, 21
 - deterministic, 21
 - ecological, 22
 - nested, 22
 - social, 22
 - System 1/System 2, 19–20
 - work on, 24–25
- Systems Assessment workshop, 323–329
 - current context, 324
 - goals of, 322, 323
 - next steps, 325–329
 - overview of, 323–324
 - Reflection canvas, 324–325
- systems team coaching, 315–329
 - description of, 315–317
 - example questions for, 321

insights from, 317
 patterns of interactions in, 317–321
 system assessments
 objectives of, 322
 Systems Assessment workshop, 323–329
 trust, 329–335
 conflicts and disagreements, 331–332,
 392–400
 example of, 400–401
 feedback loops, 332–335
 manipulative insincerity, 401
 obnoxious aggression, 401
 ORID method for, 330, 386–391
 overview of, 329–330
 practicing, 402
 radical candor, 330–331, 400–402
 ruinous empathy, 401
 Trust canvas, 391–392
 trust constellations, 392
 undesired behavior, 322
 systems thinking, 7
 application of
 Causal Loop Diagram (CLD), 31–34, 35
 doom loop technique for, 35–36
 local versus whole optimizations, 37–38
 shared responsibility in, 34–35
 collections, 21
 creative tension, 30–31
 definition of, 19
 iceberg of, 27–29
 interventions in, 37–38
 linear thinking versus, 23
 in organizational coaching, 120
 immediate improvement goals, 122
 perfection vision, 123–124
 secondary concerns, 122
 system optimization goals, 120–121
 organizational learning, 30–31
 problems solved by, 20–21
 situation analysis, 29–30
 system archetypes, 38–48
 definition of, 38–39
 “drifting goals” 46–48
 “fixes that backfire” 39–41

“limits to growth” 41–43
 “shifting the burden” 43–46
 systems
 animate, 21
 boundaries, 24–25
 definition of, 21
 deterministic, 21
 ecological, 22
 nested, 22
 social, 22
 System 1/System 2, 19–20
 work on, 25–27
 when to apply, 20–21

T

Taylor, Frederick Winslow, 9
 Taylorism, 26
 teach back technique, PBR (Product Backlog Refinement) with, 185
 team lift-off, 252, 270–277
 as foundation for success, 271
 overview of, 270–271
 role of Scrum Master in, 271
 shared purpose in, 271–273
 alignment on shared values/behavior,
 273–277
 questions to ask, 271
 team’s actions and expected outputs, 272
 team’s measures for success, 273
 team performance, measures of, 346
 team toxins, 394–397
 contempt, 394
 criticism, 394
 defensiveness, 395
 stonewalling, 395
 Team Toxins exercise, 395–397
 team work packages, 10–11
 teams
 advantages of, 13
 Axes of Change model applied to, 157–158
 building with Scrum patterns, 361–362
 coaching, 111. *See also* multifunctional learning, coaching of; systems team coaching
 conditions for team effectiveness, 291–292

- individual and team performance, 297–298
- multifunctional learning, 298–315
- overview of, 296
- role of leadership in, 295–296
- social dynamics in, 292–295
- systems team coaching, 315–329
- collaboration of Product Owners with, 355–356
- component specialized, 244–245
- cross-functional line managers, 98–99
- cross-functionality of, 257
- definition of, 12
- effectiveness of, 291–292
- expert leads, 99
- feature, 13
- feature specialized, 244
- lift-off workshops, 252
- measuring performance of, 346
- micromanagement, avoidance of, 355
- POs (Product Owners)
 - committees, 342
 - customer understanding, 347–351
 - definition of, 337
 - inward- versus outward-focused, 342–344
 - measures, 345–347, 387–390
 - Product-Part Owners, 338–339
 - promoting to team members, 340
 - in senior positions, 341
 - workshop for, 343–344
- processes and people practices in, 15–16
- product group leads, 98
- Self-Designing Team workshop, 252
- social dynamics of, 292–295
 - commitment, 293
 - constructive conflict, 293
 - integrity and trust, 293–294
 - respect, 294
 - self-control, 294–295
- stability of, 354
- structural characteristics of, 13
- team toxins
 - contempt, 394
 - criticism, 394
 - defensiveness, 395
 - stonewalling, 395
 - Team Toxins exercise, 395–397
- teams first approach, 198–199
- user connection with, 356
- work groups versus, 12
- Teams That Finish Early Accelerate Faster pattern, 314
- technical debt, 33–34, 147
- temporarily necessary waste, 53
- temporary coordination roles, creating, 91–92
- tension, creative, 30–31
- Testable Improvements pattern, 310
- test-driven development (TDD), 151
- Thinking, Fast and Slow* (Kahneman), 19–20
- thinking brain, 19–20
- This Is Lean* (Modig), 63
- Thomas, Kenneth, 393
- Thompson, James D.85, 86
- time delay, CLD (Causal Loop Diagram), 32
- time-to-market, 29
- top-down leadership, 108
- Toyota, 303, 380. *See also* Kanban
- The Toyota Way* (Liker), 73
- tracking ideas, 170
- transaction costs, 53
- transparency, 391
- Traveler role, 91
- troubleshooting
 - Agile organization design, 6–7
 - with systems thinking, 20–21
- trust, improvement of, 293–294, 329–335
 - conflicts and disagreements, 331–332, 392–400
 - conflict coaching, 398–400
 - conflict protocols, 397–398
 - guidelines for resolving, 399
 - strategies for, 397–399
 - team toxins, 394–397
 - What Color Is the Conflict? exercise, 392
 - deliberative practice, 334
 - feedback loops, 332–335
 - actions, 334
 - goals, 333
 - measures driving learning, 334–335
 - overview of, 332–333
 - ORID method for, 330, 386–391

- checklist for, 392–393
- framework questions, 387–389
- phases of, 386–387
- rational and experiential aims, 389
- summary of, 391
- overview of, 329–330
- radical candor, 330–331, 400–402
 - example of, 400–401
 - manipulative insincerity, 401
 - obnoxious aggression, 401
 - practicing, 402
 - ruinous empathy, 401
- shared values/behavior and, 273–274
- Trust canvas, 391–392
 - facilitation of, 391–392
 - overview of, 391
- trust constellations, 392
- Trust canvas, 391–392
 - facilitation of, 391–392
 - overview of, 391
- truth curve, 349–350
- Tuckman model, 271
- “two legs” rule, 313

U

- unanimity, consensus versus, 278
- unanimous decisions, 280
- uncertainty
 - Agile approach to, 3
 - types of, 348
 - uncertain dependency, 95
- underutilization, 287, 315
- undesired behavior, 322
- unit arrival rates, 54
- unit departure rates, 54
- unit functions
 - decoupling of, 76–79
 - merging with interdependencies, 79–81
- units of work, 54
- User Experience Fishbowl, 275
- User Story Mapping* (Patton), 239
- users
 - connection with teams, 356
 - role in PBR (Product Backlog Refinement), 185
- utilization
 - batch size and cycle time and, 58–59
 - variability and, 55–56
- utilization, resource, 55–56
- UX Fishbowl, 260, 284

V

- value, optimizing, 302–303
- value area heat maps, 242–249
 - component specialized team, 244–245
 - feature specialized team, 244
 - optimal starting configuration, 245
 - organizational design to eliminate dependencies, 242
 - product features, 243
 - results of, 248–249
 - value areas in, 245–248
- Value Proposition canvas, 385
- Value Stream Mapping* (Martin and Osterling), 209
- value stream mapping (VSM), 65, 209–212, 361.
 - See also* value area heat maps
 - metrics for, 210–211
 - overview of, 209–210
 - VSM workshop, 209–212
 - WIP (work-in-progress), 211
- values
 - shared
 - alignment on, 273–277
 - clarification of, 276
 - identification of, 275
 - list of core human values, 275
 - value uncertainty, 348
- variability, utilization and cycle time and, 55–56
- variables, CLD (Causal Loop Diagram), 32
- velocity, component versus feature, 258–259
- Verheyen, Gunther, 276
- vision of future, creation of, 221–233
 - importance of, 221
 - Perfection Vision workshop, 231–233
 - overview of, 231–232
 - reasons for using, 232
 - running, 232–233
 - Shared Objectives workshop, 222–227
 - examples of, 222–227

- overview of, 222
- running, 222–227
- Shared Optimizing Goals workshop, 227–231
 - closing, 231
 - developing diagram for, 229–231
 - structuring invitation for, 228
- Shared Team Expectations workshop, 227
- visualization, of work flow, 307–308
- Vodde, Bas, 91
- volatility, Agile approach to, 3
- voluntary participation, 139–141
 - characteristics of, 141
 - importance of, 139
 - lack of volunteers, 141
 - nonviolent education, 143
 - right/wrong people in, 140–141
- VSM (value stream mapping), 209–212
 - metrics for, 210–211
 - overview of, 209–210
 - VSM workshop, 209–212
 - WIP (work-in-progress), 211
- VUCA (volatility, uncertainty, complexity, and ambiguity), 3–4

W-X-Y-Z

- waiting time (WT), 54
- waste
 - behavior of queues and, 54
 - pure, 53
 - in queueing theory, 57
 - queues as cause of, 57
 - temporarily necessary, 53
- weighted SWOT analysis, 214–215
- What Color Is the Conflict? exercise, 392
- What? So What? Now What? format, 151, 267, 310, 388, 392
- Whiners, 171
- whole optimizations, 37–38
- Who/What/When exercise, 176
- WIP (work-in-progress), 24, 54, 211
- wishes, optimization goals versus, 5–6
- work
 - separation of the head from the hands, 9
 - specialization of, 9
 - work flow, visualization of, 307–308
- work groups, 12
- work-in-progress (WIP), 24, 54
- workshops
 - CLDs (Causal Loop Diagrams), 218–219
 - confrontation matrix, 214–215
 - Define Your Product, 239–241
 - materials needed, 239
 - running, 239–241
 - structural invitation for, 239
 - Definition of Done, 259–260
 - participants, 259
 - running, 260
 - structuring invitation for, 260
 - FTAM (Feature Team Adoption Map), 262–265
 - participants, 262
 - running, 262–265
 - structuring invitation for, 262
 - Gather Data, 369–370
 - Identify Patterns, 371–372
 - Market of Skills, 266–267
 - measure-for-improvement, 387–390
 - materials needed, 382
 - Measures for Learning canvas, 383–386
 - structuring invitation for, 382
 - ORID, 330, 386–391
 - checklist for, 392–393
 - framework questions, 387–389
 - phases of, 386–387
 - rational and experiential aims, 389
 - summary of, 391
 - Pattern Sequence, 365–369
 - application of patterns, 368–369
 - creation of pattern sequence, 366–368
 - materials needed, 365
 - online resources for, 369
 - selection of patterns to try, 365–366
 - structural invitation for, 365
 - Perfection Vision, 123–124, 231–233
 - overview of, 231–232
 - reasons for using, 232
 - running, 232–233
 - PO (Product Owner) activities, 343–344

- Scott Morton, 213–214
 - Self-Designing Team, 252, 265–270
 - Market of Skills workshop as preparation for, 266–267
 - overview of, 265–266
 - preparations, 267–269
 - running, 269–270
 - structuring invitation for, 269
 - Shared Objectives, 222–227
 - examples of, 222–227
 - overview of, 222
 - running, 222–227
 - Shared Optimizing Goals, 227–231
 - closing, 231
 - developing diagram for, 229–231
 - structuring invitation for, 228
 - Shared Team Expectations, 227
 - story splitting, 378–383
 - system structures, 216–218
 - materials needed, 216
 - structural invitation for, 216–218
 - Systems Assessment, 323–329
 - current context, 324
 - goals of, 322, 323
 - next steps, 325–329
 - overview of, 323–324
 - Reflection canvas, 324–325
 - team lift-off, 252
 - VSM (value stream mapping), 209–212
 - workshop design, 172–173
 - engagement and flow in, 172–177
 - structure of, 177–178
 - workshop plans, 168
 - World Cafe technique, 277
 - World Café technique, 36
 - Worren, Nicolay, 76
 - WT (waiting time), 54
- Zoom, 151