

**7 RULES**  
**FOR POSITIVE,**  
**PRODUCTIVE**  
**CHANGE**

**MICRO SHIFTS,  
MACRO RESULTS**

**ESTHER DERBY**

*Coauthor of Agile Retrospectives*

## **Praise for *7 Rules for Positive, Productive Change***

“Esther Derby has written a must-read guide for anyone whose organization is experiencing a complex shift. . . . Read this book—and learn from one of the best.”

—**Howard Sublett, Chief Product Owner, Scrum Alliance**

“This book is a product of Esther’s hard-won insights and her ability to explain them in simple, memorable ways. It is an invaluable resource to all those in the field of knowledge work who want to understand what is going on at a deeper level and how to create effective change around them. It’s an opportunity to stand on her shoulders.”

—**Kevin Trethewey, Director of Engineering, Jemstep by Invesco**

“This book is a blueprint for both novices and experienced change professionals to enhance their approach to complex change. The blend of both examples and detailed material helped me see where I need to polish my approach and where I need to improve my empathy. Selfishly, I plan to share *7 Rules* with my team and leaders so they can gain a better perspective on complex change management.”

—**Ben van Glabbeek, Vice President, Agile Transformation, Fiserv**

“Wow. If you want to help people, and the organizations they’re in, improve, these are indeed seven rules you need to know and will want to follow. It’s dangerous to go it alone. Take this book with you!”

—**Ron Jeffries, author of *The Nature of Software Development***

“Esther brings her vast experience of closely studying organizational change to show how embracing the human side of organizations means accepting them as the organic ‘forests’ they are, rather than mechanistic ‘machines.’ Her unique style of engaging storytelling and ability to carve out deep insights from everyday incidents revolving around change make this book a must-have guide of our times.”

—**Rashina Hoda, Senior Lecturer, Department of Electrical, Computer, and Software Engineering, The University of Auckland**

“An accessible yet challenging addition to the growing literature on change. I’m especially touched by the deep humanness of the approach, including the repeated reframing of situations often framed as ‘obstacles to change’ as valuable resources and opportunities to learn. Instead of merely giving lip service to complexity, Derby’s *7 Rules* embraces it.”

—**Simon Bennett, Managing Principal, LASTing Benefits (UK and Australia)**

*This page intentionally left blank*

**7 RULES**  
**FOR POSITIVE,**  
**PRODUCTIVE**  
**CHANGE**

*This page intentionally left blank*

# 7 RULES FOR POSITIVE, PRODUCTIVE CHANGE

Micro Shifts, Macro Results

Esther Derby



BK®

Berrett-Koehler Publishers, Inc.

## 7 Rules for Positive, Productive Change

Copyright © 2019 by Esther Derby

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed “Attention: Permissions Coordinator,” at the address below.



Berrett-Koehler Publishers, Inc.

1333 Broadway, Suite 1000

Oakland, CA 94612-1921

Tel: (510) 817-2277, Fax: (510) 817-2278

**BK**

[www.bkconnection.com](http://www.bkconnection.com)

### Ordering information for print editions

*Quantity sales.* Special discounts are available on quantity purchases by corporations, associations, and others. For details, contact the “Special Sales Department” at the Berrett-Koehler address above.

*Individual sales.* Berrett-Koehler publications are available through most bookstores. They can also be ordered directly from Berrett-Koehler: Tel: (800) 929-2929; Fax: (802) 864-7626; [www.bkconnection.com](http://www.bkconnection.com)

*Orders for college textbook/course adoption use.* Please contact Berrett-Koehler: Tel: (800) 929-2929; Fax: (802) 864-7626.

Distributed to the U.S. trade and internationally by Penguin Random House Publisher Services.

Berrett-Koehler and the BK logo are registered trademarks of Berrett-Koehler Publishers, Inc.

First Edition

Paperback print edition ISBN 978-1-5230-8579-8

PDF e-book ISBN 978-1-5230-8580-4

IDPF e-book ISBN 978-1-5230-8581-1

Digital audio ISBN 978-1-5230-8583-5

2019-1

Cover design by Adam Johnson. Interior design and composition by Gary Palmatier, Ideas to Images. Elizabeth von Radics, copyeditor; Mike Mollett, proofreader; Paula Durbin-Westby, indexer.

*For Jerry, whose memory is a blessing*



# Contents

**Introduction** 1

## **CHAPTER 1**

**Change by Attraction** 15

## **CHAPTER 2**

**Strive for Congruence** 31

## **CHAPTER 3**

**Honor the Past, Present, and People** 45

## **CHAPTER 4**

**Assess What Is** 63

## **CHAPTER 5**

**Attend to Networks** 81

**CHAPTER 6**

**Experiment** 97

**CHAPTER 7**

**Guide, and Allow for Variation** 113

**CHAPTER 8**

**Use Your Self** 127

**Conclusion** 141

Notes 147

References 153

Acknowledgments 159

Index 163

About the Author 173

*This page intentionally left blank*

# Introduction

PEOPLE HIRE ME BECAUSE THEY WANT DIFFERENT OUTCOMES and different relationships in their workplaces. My work almost always involves change at some level—individual, team, organization. I’ve worked with dozens of groups to understand underlying patterns and shift them toward something better.

My interest is primarily in complex changes in complex environments, working with people and systems to move toward a better outcome. I tackle problems for which there is no one obvious solution (or the seemingly obvious solution will not improve matters).

These are problems in the complex domain of the Cynefin Framework (Snowden and Boone 2007). Cynefin identifies five domains: *obvious* (until 2014 referred to as *simple*), where best practices exist and apply; *complicated*, where there is at least one right solution and expertise applies; *complex*, where new approaches must be invented; *chaotic*, where the best advice is to stabilize quickly; and *disorder*, which is a state of unclarity, not knowing which domain applies. In the obvious and complicated domains, cause and effect are evident. In the complex and chaotic domains, they can be known only retrospectively.

Snowden and Boone, in their groundbreaking article “A Leader’s Framework for Decision Making” (2007), describe the complex domain in this way:

In a complex context...right answers can’t be ferreted out. It’s like the difference between, say, a Ferrari and the Brazilian rainforest. Ferraris are complicated machines, but an expert mechanic can take one apart and reassemble it without changing a thing. The car is static, and the whole is the sum of its parts. The rainforest, on the other hand, is in constant flux—a species becomes extinct, weather patterns change, an agricultural project reroutes a water

source—and the whole is far more than the sum of its parts. This is the realm of “unknown unknowns,” and it is the domain to which much of contemporary business has shifted.

This is where I approach a situation with informed interventions to modify patterns.

I understand the appeal of best practices, prepackaged solutions, tools, methodologies, and programs that promise to solve whatever problem you have. They can be of great help for addressing problems in the simple and complicated domains. If a problem has been solved many times and there is one recognized solution, it doesn't make sense to spend time inventing something new. My research, observation, and experience tell me that, as often as not, these on-the-face reasonable solutions when applied to complex problems lead to disappointment. The situations I work with cannot be addressed directly by hiring better people, instituting a defined process, or deploying a new tool. They can be worked—I hesitate to say *solved*—only by addressing multiple and entangled influencing factors and paying attention to learning.

This book is for people like me, people who may not have “change management” in their title or job description but who nonetheless need to make changes in the way people work and the results their organizations achieve. That includes managers, executives, consultants, team leads, team members, and internal and external coaches—anyone who wants a different outcome at work.

This book is also for people who *do* have “change management” in their title, as an additional approach for organic change in complex environments.

These 7 Rules are a fine-grained addition to systemwide and human-centered large-group interventions. They apply to situations in which there isn't one right way, when it is beneficial for a group to come to their own solution, and when new solutions need to be discovered. The 7 Rules are not in themselves a process, and they aren't intended to be used in a stepwise fashion, although it is a good idea to start with Rule 1: *Strive for congruence*.

## How I Got Here

When I started my career, it was not with the ambition to revolutionize the way people approach change for their teams, departments, or entire organizations. I started my career as a programmer. I loved writing code. It was like solving puzzles, with the additional benefit of learning about different industries: light manufacturing, mortgage servicing, medical-imaging financial services, employee assistance provisioning, software as a product, and more. I didn't think much about change. My code directed a machine to manipulate ones and zeros. What did that have to do with organizational change?

Quite a bit, it turns out. I observed how my work could have a dramatic impact on others. I learned that small adjustments in the environment had the power to change patterns of work. And I witnessed the persistence of systemic patterns despite efforts to change results through incentives, training, and standardizing processes. My early experience sensitized me to notice intentional and unintentional changes and their impacts on humans and systems.

My first job was writing a program to estimate the cost of manufacturing custom decorative striping for automobiles and

to generate quotes for customers. It was fun and challenging, with the built-in opportunity to learn new things. I learned about the manufacturing process, adhesive tape stock, and the things people wanted to stick on their cars. Because my job was writing code (and, in my defense, because I was young), I didn't think much about how my program would change lives.

When the program was about to go live, I overheard the CEO of the company in a heated conversation with the chief estimator. The CEO was insisting that the estimator use the new program. I don't remember the estimator's name or what he looked like, but their conversation is still with me.

The estimator did not dispute the fact that the program produced accurate results. He just didn't want to use it. He enjoyed estimating by hand, tapping his experience and the worksheets he had developed over the years (the very worksheets that we'd used to develop the calculations). He didn't care that the program was faster and could generate many more quotes per day than he could. He didn't want to sit in front of a computer, entering data. He wanted to poke around on the manufacturing floor, talk with the people who ran the machines, converse with customers, build relationships, and leverage his years of expertise. From his point of view, what was good for the company was not good for him.

The estimator bent to the CEO's ultimatum: using the program was not as bad as losing his job. He came to work the next day, turned on his computer, and fired up the program. He worked quietly and diligently—and slowly at first. When it was time for his break, he went down to the factory floor and had a smoke with the guys. He didn't complain to them about having to use the program (perhaps out of fear of repercussions), but the fact that he had accepted the program did not go unnoticed. His example may have made it easier for other people to suspend



judgment and give automation a try. Or they may have seen that if the chief estimator couldn't stem the tide of automation, they may as well go along with the times.

## Two Lessons about Change

That conversation was a turning point for me, and it altered how I thought about my job. It also taught me two important lessons about change.

- **Any given change may be a positive for some people and a negative for others.** I didn't just write computer code: I changed the way people worked and, potentially, how they viewed their jobs and even their employment. Attending to that fact early and often can both shape an implementation and reduce anguish.
- **Change is a social process.** Change can affect relationships, status, sense of identity, and self-perception of competence and worth. Learning happens through social interaction. How other people respond to an idea influences those around them.

My next programming job was with a large financial services company, which I'll call Bradley's. It was quite a career step to go from a small, family-owned manufacturing company in an unfashionable suburb to a Fortune 100 firm with a high-rise headquarters in Manhattan and tens of thousands of employees around the globe. Bradley's record number of profitable quarters was envied in the industry, and its brand was well known. I joined a group writing code to account for stock and bond holdings. This time the people who used the programs welcomed our automation of hundreds of accounting entries associated with

daily purchases, sales, and market price updates. I turned my attention to a different sort of change.

After several months on the job, I developed some definite ideas about how to make our work as programmers more effective. One of my changes involved making data visible. I posted on the wall a big diagram that showed the sequence and dependencies of all the programs that ran every night. When a program crashed or had some problem that required a pager call, I stuck a pin in that program on the diagram. Soon we had data.

I am pretty sure that someone in a different department in a different building already had that data, but it wasn't available to us. My coworkers and I started paying more attention when changing error-prone programs. We took extra care with testing, and whenever we encountered messy code, we cleaned it up.

Another small change involved adding a printed diagram of each program's structure to the program listing binder. I didn't even have to research and diagram the structure myself; I only needed to select the setting to generate the printout. *Voilà!* This little change made it easier for new hires to grasp the flow of a program, and it made locating bugs easier, too. Plus it took much less time than speed-reading an 8-inch-thick stack of computer code for subroutine headings.

These were changes I could make without permission, budget, or persuading anyone else that it was a good idea. I made small adaptations to the environment and modeled the change I wanted to see. These very small changes often made a significant difference.

Eventually, I changed jobs again, moving to a management role at Bradley's. I no longer spent my days figuring out how to get a computer to do what needed to be done and making small changes on the side. Adjusting the environment to enable

more-productive work became my full-time job (along with budgets, reports, staff meetings, and a host of administrative details). My opportunities to learn about people and change expanded again.

## Seeing the Big Picture

As a manager I had a certain amount of control and power. And I very soon realized the limits of that power when it came to change. Telling people to do things differently didn't work. I needed to provide context and information. I needed to listen and learn. Tweaking the environment wasn't always sufficient. I needed to learn how to influence and see beyond what was happening in my own group.

One of my no-permission change projects involved working with other development managers on project estimation. We had noticed—it would have been impossible *not* to notice—that most projects in our information technology (IT) department missed their budget and scheduling commitments. And they were not just a little overbudget but 200 percent or more and well behind schedule. Project results at Bradley's were even worse than published industry statistics, which were already dismal. On some level these failures were entirely predictable, given that so few projects achieved their targets. Yet each failure produced a mini-drama with recriminations and admonitions that it must not happen again.

We analyzed as much of the data as we could find and discovered some interesting patterns. The larger the project, the more likely it was to bust the budget and blow through deadlines and the larger the discrepancy was likely to be. Projects that involved a new technology, teams who didn't have

domain knowledge, and developing new products—all were guaranteed to exceed original estimates. The projects that came closer to their original projections were either small or not very complicated, even when they involved lots of work. It wasn't the project managers who made a difference; it was the type and size of the project.

This pattern persisted across the entire department and over several years. We established categories and confidence factors based on the data. Our idea was to provide a range estimate rather than a point estimate and include a confidence factor. Our theory was that the range and confidence factor would communicate the uncertainty involved.

Our experiment was a step in the right direction, based on our knowledge and thinking at the time. We did find peers who were interested in our ideas, but we didn't get very far. Neither the budgeting process nor those who owned it would accommodate a range estimate. The finance people wanted a precise commitment, and the budgeting system could accommodate only one number.

Given our sphere of influence and lack of official standing, we recognized that we weren't likely to succeed with the budget director. Nor would we gain funding to modify the budget system to accept a range. So, we continued using the categories to manage expectations locally, and we looked for other ways to make the inevitable overruns less surprising to those who weren't close to the work.

Meanwhile our IT executives were attacking this very issue of budget and project overruns. They had access to traditional levers of power—incentives, hiring and firing, policy, process, money—and they used them. Their efforts to change