

JOY OF AGILITY



How to Solve Problems
and Succeed Sooner

JOSHUA KERIEVSKY

Praise for *Joy of Agility*

“This delightful book shines a light on both why and how to create agility in your organization. Through stories that are as engaging as they are insightful, Joshua Kerievsky shows how a culture of fear and a mindless pursuit of efficiency drive out learning and stifle agility. A set of simple, actionable mantras provide the way forward.

—Amy C Edmondson, professor at Harvard Business School and author of *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth*

“Very well crafted and told; it’s hard to imagine a more illuminating guide to this mission-critical subject.”

—Harry Beckwith, *New York Times* bestselling author of *Selling the Invisible*

“Stories influence our behavior more often and more effectively than logic, no matter how much we might pretend otherwise. Joshua Kerievsky has distilled his expertise into six mantras and illustrated each with short, captivating stories. I found the stories so engaging that I read almost the entire book in one evening—it was that difficult to put down.”

—Mary Poppendieck, coauthor of Lean Software Development series

“Joy is the perfect word for this book. Joshua takes a beautiful approach to change by sharing compelling stories of agility in so many different industries and contexts. It will be impossible to not have a dozen of these stories inspire you on your own journey to the very human *Joy of Agility*. Read it, run the experiment, and start your joyful journey of change.”

—Richard Sheridan, CEO and chief storyteller at Menlo Innovations and author of *Joy, Inc.: How We Built a Workplace People Love* and *Chief Joy Officer: How Great Leaders Lift Human Energy and Eliminate Fear*

“Joshua Kerievsky understands something very important about our species. We are storytellers. It is not only our preferred way of learning but also how our brains make sense of the world. Joshua’s stories are inspirational and will lead you to discover the real joy of agility. I love this book.”

—Linda Rising, coauthor of *Fearless Change* and *More Fearless Change*

“Joy is a tell. If you have joy on your team, you are in the most prized collaborative state. There is no better step you can take toward experiencing team joy than reading this book. It is a treasury of joyful practice, and you deserve to have it all.”

—Jim McCarthy, coauthor of *Software for Your Head* and *Dynamics of Software Development*

“Whether you’re an accomplished or aspiring agilist (or somewhere in

between), this rich compendium is a bounty for anyone who wants to foster a culture of adaptability, collaboration, and improvement, while creating a more intentional, efficient, sustainable, and—above all—outstanding work experience and product. Engaging and inspiring, *Joy of Agility* is an absolute joy from beginning to end!”

—Tonianne DeMaria, coauthor of Shingo-award-winning *Personal Kanban*

“Joshua, a fervent agile pioneer, understands the power of story and has assembled a host of stories that present facts and ideas about “being” agile as opposed to “doing” agile. Like shades of color on a palette, *Joy of Agility* enables us to understand both the broad sweep and nuances of agility; Enjoy. Learn. Remember. Act.”

—Jim Highsmith, agile pioneer, author, and former executive consultant at
Thoughtworks

“Reading *Joy of Agility* makes me want to hand this book to all of my clients. I love the definition of agile and the compelling stories that make one both think and laugh! The stories are short, crisp, and to the point. As a person who likes to highlight important things, I was inclined to highlight each and every story. This is an excellent and very insightful book!”

—Jutta Eckstein, business coach, change manager, and author of *Company-wide Agility with Beyond Budgeting, Open Space & Sociocracy, Agile Software Development in the Large*, and *Agile Software Development with Distributed Teams*

“Joshua’s masterful story curation won me over completely as a reader that doesn’t even like story-centered books. Joshua has become the consummate voice of agility; there in the early days, helping shape and deeply understand the new practices introduced by the Agile movement. He continues to refine, expand, and reinvent that understanding for a new generation of practitioners. Whether you have been practicing agility for 20+ years or have barely heard of the concept, you’ll want to pick up *Joy of Agility*.”

—Michael Spayd, founder of The Collective Edge and coauthor of *Agile Transformation: Using the Integral Agile Transformation Framework to Think and Lead Differently*

“Brilliant work and a dazzling diversity of stories!”

—Pat Reed, executive agile consultant in Australia and USA

JOY OF AGILITY

Also by Joshua Kerievsky
Refactoring to Patterns

JOY OF AGILITY
How to Solve Problems
and Succeed Sooner
JOSHUA KERIEVSKY

Matt Holt Books
An Imprint of BenBella Books, Inc.
Dallas, TX

Joy of Agility copyright © 2023 by Joshua Kerievsky

All rights reserved. No part of this book may be used or reproduced in any manner whatsoever without written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

Matt Holt is an imprint of BenBella Books, Inc.

10440 N. Central Expressway

Suite 800

Dallas, TX 75231

benbellabooks.com

Send feedback to feedback@benbellabooks.com

BenBella and *Matt Holt* are federally registered trademarks.

First E-Book Edition: 2023

Library of Congress Control Number: 2022031856

ISBN 9781637742778 (hardcover)

ISBN 9781637742785 (electronic)

Editing by Gregory Newton Brown

Copyediting by Michael Fedison

Proofreading by Isabelle Rubio and Cape Cod Compositors, Inc.

Text design and composition by PerfecType, Nashville, TN

Cover design by Brigid Pearson

Special discounts for bulk sales are available. Please contact bulkorders@benbellabooks.com.

*For
Bruce, Diana, Dmitri,
Sasha, Sophia, and Eva,
and most of all, Maria.*

Contents

[Preface](#)

[Introduction](#)

[Taking Action](#)

[The Agile Birth of a Shoe Company](#)

[The Definition of Agile](#)

[Six Agile Mantras](#)

[Break On Through to the Agile Side](#)

[You Can Still Dunk in the Dark](#)

[Chris Rock's Little Bets](#)

[Let There Be Water Balloons](#)

[Banking On Joy](#)

[Eva Learns to Ride](#)

[Consider Your Options](#)

[Airbnb's Shipping Culture](#)

[Oprah Winfrey in Texas](#)

[Investing in Agility at American Airlines](#)

[A Bar, a Blender, and a Cheesesteak](#)

[Be Quick—But Don't Hurry](#)

[The Wooden Way](#)

[Take the Easy Way, It Gets Harder; Take the Hard Way, It Gets Easier](#)

[Slack, a Prescription for Increased Agility](#)

[Fine Products, Crafted Efficiently](#)

[Stop Starting, Start Finishing](#)

[Align to Accelerate](#)

[Maximize the Amount of Work *Not* Done](#)

[Fast, Frugal Feedback from a Feature Fake](#)

[Speed Up by Taking Smaller, Safer Steps](#)

[Size Teams for Few to No Slow Handoffs](#)

[Doing the Impossible Fifty Times a Day](#)

[Haste Makes Waste](#)

[Don't Make People Wait](#)

[Ward's WikiWikiWeb](#)

[Despots and Dentures](#)

[Don't Converge Prematurely](#)

[Fail Fast \(But Don't Rush\)](#)

[Summary: Be Quick—But Don't Hurry](#)

Be Balanced and Graceful

[Habitual Excellence](#)

[Mixed Messages](#)

[Gary V's ROI of Caring](#)

[Balanced Teams Perform Better](#)

[Awful to Awesome at Yahoo!](#)

[Balanced Teams Are CLEAR](#)

[Stop Interrupting People](#)

[Etsy's Blameless Culture](#)

[The Blame Game](#)

[Discover Art in Mistakes](#)

[Give Feedback Gracefully](#)

[The Emperor's New Table](#)

[When the Plans Came Tumbling Down](#)

[From Sith Lord to Superboss](#)

[Change by Attraction](#)

[Summary: Be Balanced and Graceful](#)

Be Poised to Adapt

[Virginia's Impromptu Solo on the D String](#)

[How NASA Prepared for the Unexpected](#)

[A Personalized Purchase](#)

[Multiplicity and Selection](#)

[The *Gossamer Condor*](#)

[Iterative Experimentation](#)

[Fail Smarter via Quick, Cheap Experiments](#)

[How Jeff Bezos Treats Failure and Success](#)

[Innovating Without Permission at Amazon](#)

[Product Ownership Is a Team Sport](#)

[Safety and Skill in a Rapid Pivot](#)

[Learn Sooner](#)

[Are They Using It?](#)

[Usability Evaluations for Rapid Learning](#)

[The Day We Stopped Sprinting](#)

[Summary: Be Poised to Adapt](#)

Start Minimal and Evolve

[Cheap, Minimal Teleportation](#)

[Deliver Value Sooner](#)
[An Evolutionary Path](#)
[Begin with a Primitive Whole](#)
[Birth of an Online Shoe Company](#)
[The Ballroom](#)
[Sketch, Craft, Refine](#)
[Evolving a New Ticketing System](#)
[Birthing a Best Seller](#)
[Summary: Start Minimal and Evolve](#)

Drive Out Fear

[Make Safety a Prerequisite](#)
[Safety Is Golden](#)
[Robert Plant and My Ticket Ordering Ordeal](#)
[Pixar's Near Miss](#)
[The Shocking Truth About Those Brown M&M's](#)
[Amy Edmondson's Eureka Moment](#)
[Is It Safe to Fail?](#)
[Never Lose Your Shield!](#)
[A Virgin Derailment](#)
[Heroism Is Hazardous](#)
[If It Ain't Broke, De-Risk It](#)
[Pre-Work Hazard Analysis](#)
[Healthy Fear](#)
[Be Careful \(But Don't Play It Safe\)](#)
[Silence versus Voice](#)
[The Cinder Block Incident](#)
[The Cycle of Mistrust](#)
[Norm Kerth's Safety Poll](#)
[Resiliency via Diversity](#)
[Stop Work Authority](#)
[Become Resilient](#)
[Summary: Drive Out Fear](#)

Be Readily Resourceful

[Birth of an Airline](#)
[A Nordstrom Ninja and a Raving Fan](#)
[Readily Resourceful in a Recession](#)
[The Kondo Way](#)
[Bullet-Train Thinking](#)

[Questions Are the Answer](#)

[The Wright Stuff](#)

[It Was All Greek to Me](#)

[Take Personal Responsibility for Learning](#)

[Business as Usual](#)

[Changing Our Perspective on Change](#)

[Get Comfortable with Being Uncomfortable](#)

[Learned Helplessness](#)

[Be Resourceful, Not Resentful](#)

[Got a Coach?](#)

[Engage an Ensemble](#)

[How Pete Sampras Won Big by Learning to Lose](#)

[The Change Curve: Virginia Satir's Six Stages of Change](#)

[The Change Curve for My Tennis Serve](#)

[Celebrate Chaos](#)

[Make Problems Visible](#)

[Halloween in April](#)

[Tear Down the Cubicles](#)

[Fearless Change](#)

[Summary: Be Readily Resourceful](#)

[In Conclusion](#)

[Acknowledgments](#)

[Bibliography](#)

Preface

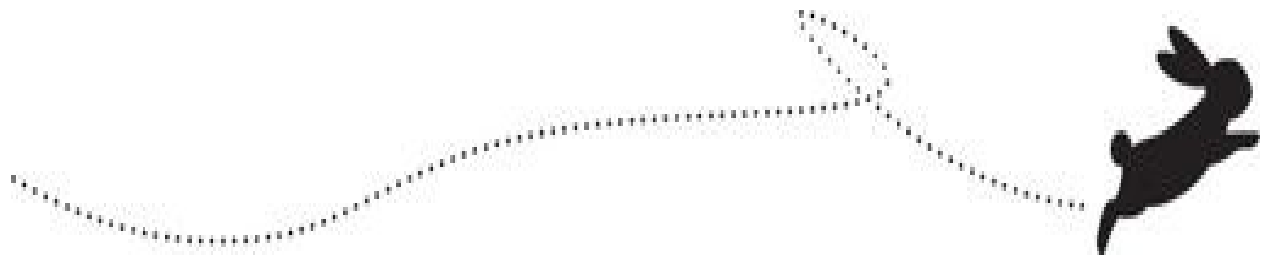
One of the greatest gifts of my life has been discovering how agility enables excellence. I love finding ways to make slow and awkward things faster and easier, solving difficult problems by being readily resourceful, and adapting to change with speed and grace. It's the joy of agility.

In the late 1990s, I was part of a small movement that pioneered what would become agile software development. We experimented with ways to remove anything that slowed us down or made work difficult. My interactions with colleagues became smoother, collaboration with customers became closer, feedback came faster, making changes became safer, and handling unexpected events became easier. We learned how to sustainably create more value with less waste. That ultimately helped us produce happier customers sooner.

Several decades spent experimenting, learning, unlearning, and reflecting on the nature of agility—including helping my own company and our clients become more agile, applying agility to home repairs, health, fitness, parenting, and more—led to this book.

Joy of Agility begins by tackling the widespread confusion surrounding the definition of the word *agile*. It then presents six agile mantras that help make agility actionable. Finally, the majority of the book is an anthology of stories. These stories serve to clarify one or more agile mantras and come from many professions, including business, education, entertainment, sports, psychology, marketing, aviation, and more. Each story concludes with a call to action.

I will be delighted if even one of these stories inspires you to put this book down and take action.



Introduction



Taking Action

One morning in 2002, more than five years after starting my own company, I read a book that revealed a huge mistake I was making. The book *Selling the Invisible: A Field Guide to Modern Marketing* by Harry Beckwith is a collection of insightful stories, each of which concludes with a call to action.

That morning, two of the stories I read had such an impact on me that I immediately changed how I was marketing my company.

The first story, called “Fanatical Focus,” explained the importance of not trying to be all things to all people. Instead of focusing on quality, price, and value, the pizza company Domino’s relentlessly stressed its speed: “30 Minutes or It’s on Us.” That story concluded with the call to action: *Stand for one distinctive thing that will give you a competitive advantage.*

The second story, called “The Fear of Positioning,” amplified the first story so intensely that I had to put the book down to think about how I could apply what I had just learned to my own business. It explained how most companies are afraid to stand for one thing since that means not standing for a bunch of other important things.

In this story, the author explains how Scandinavian Airlines (SAS) was struggling badly in 1980. Faced with a \$20 million loss, the airline decided to reposition itself as the “Business Traveler’s Airline.” Despite major concerns that they would lose the business of its bread-and-butter tourist travelers, SAS forged ahead with the idea, offering a new service called “Euro-Class.” Business travelers could now get “olives in martinis, bigger seats, phones, telexes, a separate four-minute-faster check-in counter, and free drinks, newspapers, and magazines.”

SAS made \$80 million the first year it offered Euro-Class. And what about the tourist passengers, seemingly abandoned by this new position? They were happier than ever! Given all of the sales of expensive business-class seats, SAS could afford to offer bargain-basement prices for economy seats, which tourists snapped up. Flights were full, and “standing for one thing” was the secret behind SAS’s success. The call to action at the end of this story was simple and direct: *To broaden your appeal, narrow your position.*

After reading that story, I considered my own company’s position. The home page of our website offered five different software development services. We were most definitely not standing for one thing! At the time, Design Patterns was our cash cow, but Extreme Programming (XP) was a bigger and more important service. I decided right then and there to reposition my company as *the XP*

company and immediately changed our website to make the new position clear.

In the months and years that followed, Industrial Logic became known around the globe as the source for the world's best XP training and coaching. Revenue climbed, the company grew, and we even continued to have plenty of Design Patterns business. All of that happened because I'd read a couple of Harry Beckwith's brilliant stories.

The stories in *Joy of Agility* are meant to inspire *you* to take action. As you read each one, ask yourself how you would apply the lesson to your context.

Joy of Agility is also a springboard to other books. Throughout my career, I've learned a great deal from brilliant authors. You'll find references to their books scattered throughout the chapters that follow and in the bibliography at the end of this book. Each one is an invitation to further explore aspects of agility.



The Agile Birth of a Shoe Company

When Kenneth Cole founded his footwear company in 1982, the odds were stacked against him. His funds were limited, and the U.S. economy was in recession. To manufacture his shoes, he needed money, but obtaining a bank loan during a tough economic period would be difficult. To market and sell his shoes later that year, he would need footwear buyers and sellers to see his products at the New York Shoe Expo, which took place at a Hilton Hotel in Midtown Manhattan. However, unlike established shoe companies, he couldn't afford the expensive showcase rooms at the Hilton.

Like Kenneth Cole, we all face challenges. Regardless of our profession, we struggle to find our way through tough conditions, unexpected events disrupt us, new obstacles impede our progress, and the status quo keeps pulling us back. Organizations typically struggle with too much bureaucracy, insufficient innovation, poor collaboration, serious quality problems, disempowered and unmotivated staff, and ever-increasing pressure to deal with nimble competitors.

What could help us overcome such challenges?

Being agile.

Agility helps us solve problems with quick, easy grace. As you'll learn later in this book, it helped American Airlines rapidly respond to a travel mandate during the beginning of the COVID-19 pandemic. It helped Oprah Winfrey successfully defend herself from a lawsuit while continuing to air her hit show. It helped Paul O'Neill rescue a one-hundred-year-old industrial giant. It helped my own company pivot successfully to retain our business with Google during a global recession. And it helped Paul MacCready's team invent the world's first human-powered airplane.

It also helped Kenneth Cole overcome the challenges he faced when he started his shoe company. His first decision was to avoid the delays and difficulties of seeking a bank loan to fund his line of elegant women's shoes. Instead, he flew to Italy and found Italian shoemakers who needed his business and would manufacture his shoes on credit.

Returning to New York with thousands of pairs of shoes, he had to find an affordable and effective way to showcase his products at the New York Shoe Expo. On a whim, he called a friend in the trucking business to see if he could park a forty-foot trailer across the street from the Hilton. His friend was happy to lend him the trailer but said that parking it in Midtown Manhattan for several

days would be impossible. Undeterred, Kenneth called the mayor's office to learn about the city's parking rules. He discovered that only utility or motion picture production companies could park large vehicles on Manhattan streets for extended periods of time.

That afternoon, Kenneth bought stationery and business cards with a new company name, Kenneth Cole Productions, Inc. The following morning, he filed for a permit to shoot a feature-length film called *The Birth of a Shoe Company*. It was quickly approved. Kenneth then decorated the trailer, hired actresses to model the shoes, and brought on a director.



When the New York Shoe Expo began, hundreds of curious shoe buyers lined up to explore what was inside the trailer, while the director filmed the action. Within two-and-a-half days, Kenneth Cole sold forty thousand pairs of shoes.

Being agile helped Kenneth Cole succeed. When he launched his shoe company, his business dealings were quick; he adapted to tough economic conditions with ease; he was resourceful in finding a unique and inexpensive way to stand out from the crowd; and he was graceful in the creative way he sold

thousands of pairs of shoes.

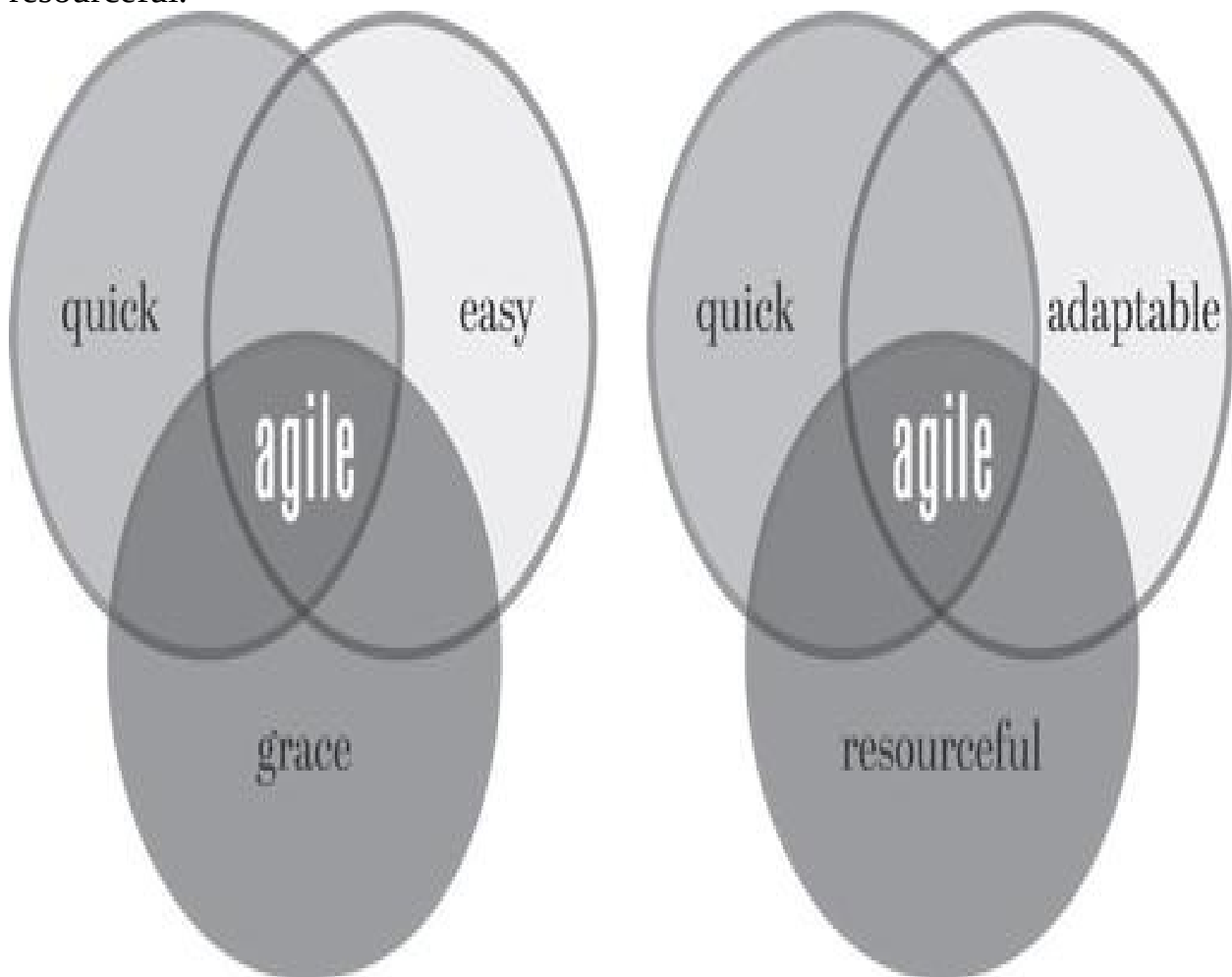
Kenneth Cole's story, and others like it in this book, is meant to help you understand what is required to be agile. Before we dive into those stories, let's first review what the word *agile* means.



The Definition of Agile

Agile is an adjective that further defines a noun: e.g., an *agile* dancer, driver, pilot, nurse, programmer, executive, manager, team, or department. The Merriam-Webster dictionary defines agile as “marked by ready ability to move with *quick easy grace*” or “having a *quick resourceful and adaptable* character.” Both of these definitions are useful for understanding how to be agile in your context.

Being agile involves—yet isn’t exclusively about—speed; hence the word *quick* in both of Merriam-Webster’s definitions. Being quick does not mean hurrying, rushing, or otherwise moving so fast that you are out of control. And let me repeat: *being quick does not mean being agile*. As you can see in the two Venn diagrams below, it also requires ease and grace or being adaptable and resourceful.



Saying someone is “quick and agile” is redundant, since being agile requires being quick. And finally, we often need to slow down in order to learn how to be

genuinely quick.

Being agile involves *ease*. When you witness agility, it often looks effortless. Ease in movement or bearing often results from great skill and practice.

Agility also requires *grace*, which in this context means “ease and suppleness of movement or bearing.” Suppleness associates grace with adaptability, since the word means “readily adaptable or responsive to new situations.” When we are graceful, we interact harmoniously with others, we move with ease, we are balanced, relaxed, and able to respond to new situations without hesitation.

Putting all of that together, being agile requires movement or behavior that is at once quick, effortless, balanced, and harmonious. It isn’t easy!

When we are agile, we are readily *resourceful*, able to quickly, creatively, and skillfully handle difficulties or new situations.

Being agile also involves being *adaptable*, capable of adjusting to new conditions or uses. Adaptability isn’t synonymous with agility. If it takes you a long time to adapt to something, you aren’t agile. Agility requires being quickly and easily adaptable.

Putting all of that together, being agile requires being quick and clever in overcoming or adapting to new or unexpected situations.

Fear stifles agility. When we are afraid of being injured, blamed, penalized, or dismissed, we become hesitant, defensive, or even paralyzed, and our ability to be quick, graceful, adaptable, and resourceful is diminished.

Becoming agile requires ongoing practice. It’s a journey, not something you learn quickly in a book, article, or class. No one becomes agile merely by acquiring a certification in someone else’s idea of an agile process. Just as it takes dedication and practice to gain skill in a musical instrument, writing, cooking, or a sport, you will also need to study and practice agility in order to gain skill in it. Agile mantras, the subject of the next section, provide a way to deliberately practice agility.



Six Agile Mantras

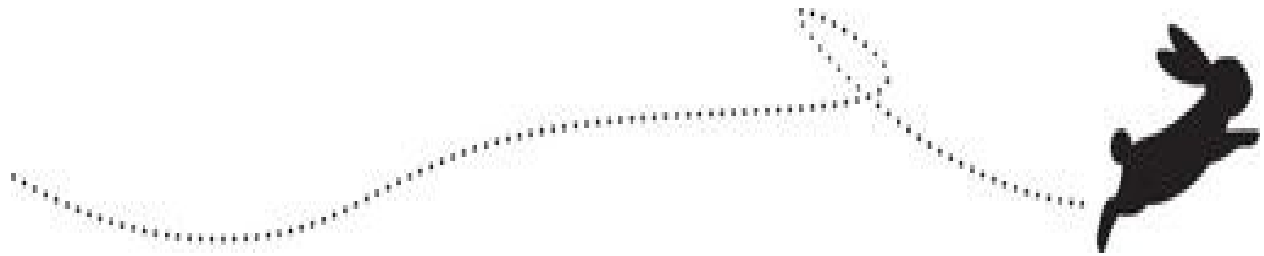
A mantra is something you contemplate regularly and that continuously guides your decisions and actions. The six agile mantras in this book stand on the shoulders of giants in fields as diverse as quality management, athletic coaching, resilience engineering, psychological safety, and business leadership. These mantras exist to help individuals, groups, teams, departments, divisions, and whole organizations develop agility.

The six agile mantras also form the major parts of this book. In each part, you'll find a description of the mantra along with numerous stories that further explain it. For now, let's look at brief summaries of each of the six mantras:

- *Be Quick—But Don't Hurry*: Succeed sooner via quickness under control and avoid costly mistakes from hurrying or rushing.
- *Be Balanced and Graceful*: Resolve imbalances, collaborate harmoniously with others, and make change empowering, not forced.
- *Be Poised to Adapt*: Develop a readiness to revise or respond quickly and easily to change.
- *Start Minimal and Evolve*: Begin quickly and easily with something basic and rapidly improve it based on what you learn.
- *Drive Out Fear*: Make safety a prerequisite to protect people and pave the way for high performance.
- *Be Readily Resourceful*: Solve problems without hesitation by being quick, clever, and creative.

The bulk of this book is devoted to helping you understand each of the six agile mantras via stories. Each story explores the nuances of a mantra, so you can more deeply understand what it means. Since all six of the agile mantras are closely related, you'll often find two or more agile mantras present in any given story.

Before we explore each agile mantra, I first want to share a few stories that exemplify agility and illustrate the power of simultaneously applying many agile mantras at once.



Break On Through to the Agile Side

The stories here are some of my favorites for explaining what it means to be agile. Each describes a breakthrough resulting in a joyful outcome to a challenge.

As you read each story, notice the decisions and behavior of the people, teams, or organizations. Then ask yourself which of the six agile mantras are present. Doing this will help you get better acquainted with agility and the six agile mantras before we unpack each mantra in more depth.



You Can Still Dunk in the Dark

On February 3, 2013, a couple of minutes into the third quarter of the Super Bowl, the Superdome in New Orleans was suddenly plunged into almost complete darkness. It was an astonishing moment during America's biggest sporting and advertising event of the year. With the entire stadium in near-complete darkness, the championship game between the Baltimore Ravens and San Francisco 49ers came to an abrupt halt.

What happened? Was it a terrorist attack? A post-Hurricane Katrina breakdown in critical infrastructure? No one knew during the first few eerie minutes.

Gradually it became clear that a partial power outage had occurred. It took twenty-two minutes to restore power, and play resumed thirty-four minutes after it had been halted.

During the blackout, a beautiful example of team agility occurred, but the team I'm talking about wasn't on the football field—it was in an office 1,300 miles away. Soon after the stadium went dark, a thirteen-person group working for Oreo, America's best-selling cookie, seized the moment to compose and publish the tweet on the following page.

The tweet went viral. Oreo's follower count on Twitter, Facebook, and Instagram grew exponentially. Accolades poured in via social media and email. The following day, the advertising industry declared Oreo the winner of the Super Bowl. One media outlet announced that the best ad of the Super Bowl wasn't an ad; it was a tweet. Oreo wanted to remind everyone that it was America's favorite cookie, and its agile branding and marketing team had succeeded wildly in achieving that goal.

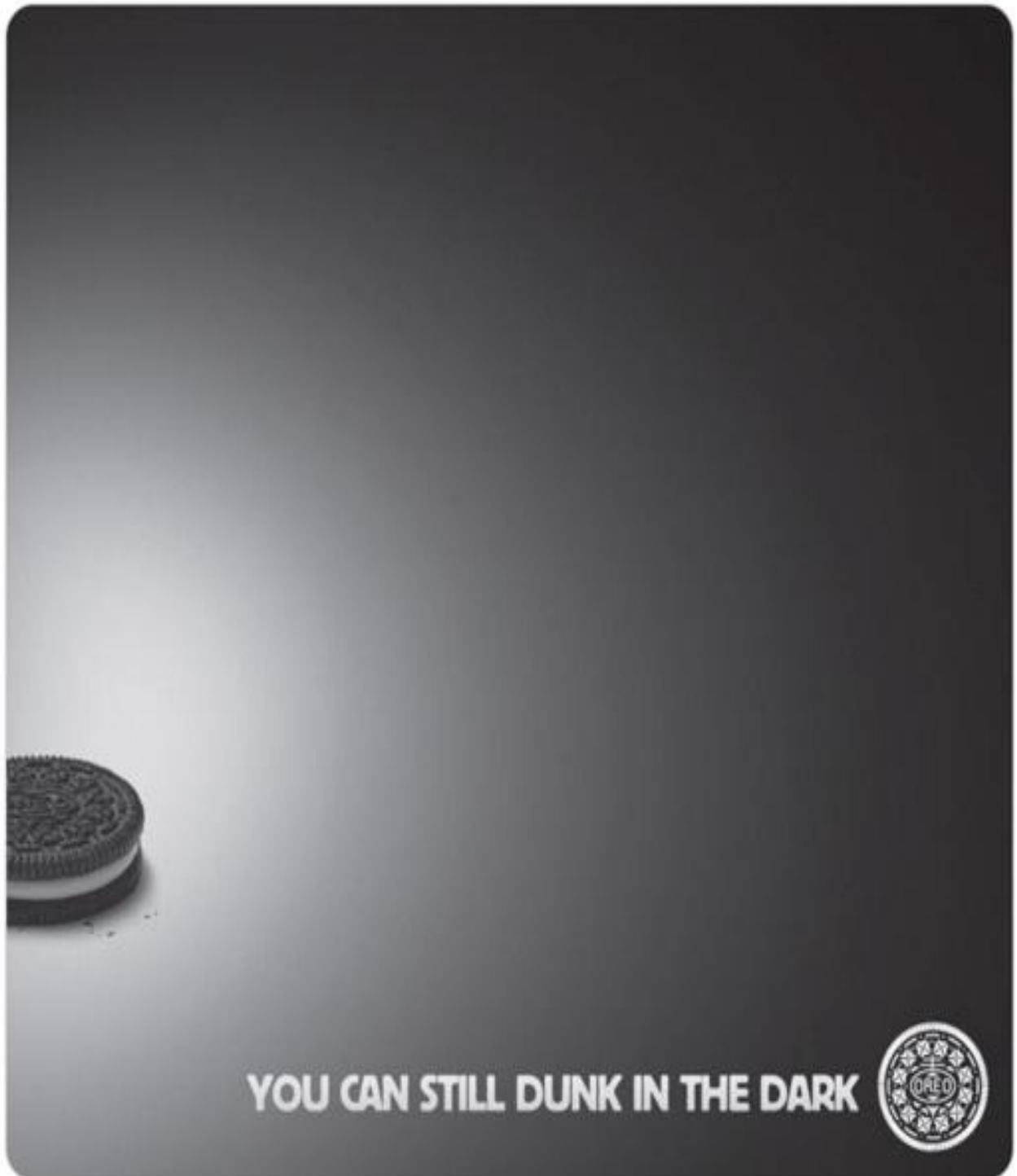
How was this epic tweet created?



OREO Cookie ✓
@Oreo

...

Power out? No problem.



You might think that a few self-organizing Oreo creatives suddenly saw an opportunity, hurried to capitalize on it, and rushed out the tweet. Yes, the Oreo advertising team saw an opportunity and acted on it, but that isn't the full story of the famous blackout tweet. The story was actually years in the making.

Oreo turned one hundred years old in 2012. To celebrate its birthday, the company wanted to show just how deeply ingrained the Oreo cookie was in American culture. For one hundred consecutive days, Oreo posted social media content associating the cookie with anything noteworthy in the news, such as the Mars rover landing, Elvis week, and the anniversary of the creation of the rainbow gay pride flag. Called the "Daily Twist," members of Oreo's brand team and a few advertising agencies collaborated daily in a room in New York City.

Working closely together in one room enabled the team to respond quickly and creatively to interesting cultural events. By the time the Super Bowl took place in 2013, the Daily Twist had helped the team develop muscle memory for real-time branding and marketing. Michael Nuzzo, creative director for the advertising agency 360i, recalled, "We had these daily content calendars across social platforms and every day we would review the content calendar. We'd been very thoughtful, planned in advance what we would do. Short of a meteor falling out of the sky, we knew."

Ten people from the team were in the war room on Super Bowl Sunday, including Michael Nuzzo; Maggie Walsh, 360i's strategist; copyeditors; graphic designers; and Oreo brand representatives. Lisa Mann, VP of Cookies for Kraft Foods (owner of Oreo at the time), and Sarah Hofstetter, 360i's CEO, watched the game from home and were connected to the team via their laptops and phones. The day before the Super Bowl, the team had met in the war room to test the technology and rehearse how they would perform on the big day. "We were overprepared," said Walsh. "We had this dry run on Saturday. And we had this worksheet: so, what happens if this person does this, this person does this, who writes the copy, who presses *send*. We had all the tracking in place."

Soon after the blackout began, Michael Nuzzo said to everyone in the war room, "We should probably do something." Seconds later, Sarah Hofstetter sent an email to the team with the subject, "power—opp to capitalize?" The team replied that they were already on it.

Nuzzo threw out the line, "You can dunk in a blackout" and in under a minute, the team had the Twitter post composed. The post needed approval from Oreo, and fortunately a high-speed approval process was already in place. Oreo's associate brand manager, Danielle Brown, also in the war room, quickly made sure that no terrorist event had occurred and that it was safe to post. Next, she emailed Lisa Mann for final approval, writing, "I really want to say, 'You can

still dunk in the dark.’” Lisa loved it and quickly gave final approval.

The post was updated and published. From start to finish, the team had taken only a few minutes to design, caption, approve, and publish their epic tweet.

So, while the Oreo creatives saw an opportunity, worked fast to capitalize on it, and quickly pushed out the tweet—providing the world with a marketing case study in what agility is all about—Lisa Mann would say that it really took two years to publish that tweet. It involved conceiving of Oreo’s branding and marketing strategy to commemorate its one hundredth birthday; forming the Daily Twist team to produce content for one hundred consecutive days; gaining skill in associating the cookie with daily newsworthy events; and ultimately growing a team that could reliably perform high-quality, real-time branding and marketing.

The epic tweet didn’t involve hurrying and wasn’t a spontaneous act of genius. It was the result of the right mix of people—a balanced team—learning how to collaborate gracefully and make decisions rapidly in order to be poised to adapt to real-time events. Without such work, the team would not have been able to be so readily resourceful in a profoundly unexpected situation—like a Super Bowl blackout.

Agility requires practice and preparation.



Chris Rock's Little Bets

I love seeing great comedians perform. The truly great ones make me cry with laughter and keep me laughing throughout their set with their insights, timing, intonation, expressions, and body language. Have you ever wondered how such performances become so funny?

In his book, *Little Bets: How Breakthrough Ideas Emerge from Small Discoveries*, Peter Sims reveals what star comedians like Chris Rock do to create comedy gold.

Sims calls it a “rigorous experimental discovery process.” Chris Rock routinely shows up unannounced at a comedy club in New Brunswick, New Jersey—about an hour’s drive from his home. The audience of fifty or so people is thrilled to have this star comedian perform for them. But after Chris Rock takes the stage, the audience soon discovers that Rock isn’t his normal funny, animated self.

Sims explains that when Chris Rock tests out new material, he doesn’t “launch into his familiar performance mode that fans describe as ‘the full preacher effect,’ when he uses animated body language, pitchy and sassy vocal intonations, and erupting facial expressions.” Instead, Rock sits calmly on a barstool with a legal pad filled with ideas and casually tests out each idea while observing his audience intently, “noticing heads nodding, shifting body language, or attentive pauses, all clues to where good ideas might reside.”¹



Developing a great new comedy set requires operating under what Eric Ries, author of *The Lean Startup*, calls “conditions of uncertainty.” To learn what is or isn’t funny, Chris Rock starts in the safety of a small, intimate venue, where he can protect his brand and experiment without fear. To discover and develop the few gems that will make audiences erupt with laughter, he takes little bets, quickly testing hundreds of raw or uneven pieces of content. Eventually, he experiments in slightly larger venues with bigger audiences. Night after night, for weeks or months, he removes uncertainty, always being poised to adapt his comedy set until it is consistently funny and ready for the biggest stages.

Under conditions of uncertainty, take little bets. Apply a rigorous experimental discovery process.

[1.](#) Peter Sims, *Little Bets: How Breakthrough Ideas Emerge from Small Discoveries* (New York: Free Press, 2011), 1–2.



Let There Be Water Balloons

A few days before she turned seven, my daughter Eva asked if she could have a water balloon fight during her birthday party. “Sure,” I immediately replied, but the thought of making more than a few water balloons filled me with dread. I knew this meant hours of awkward fiddling—first pick out an empty balloon, fill it with water, hope it didn’t pop prematurely, and then use my not-so-nimble fingers to tie the slippery, wet end into a tiny knot. Over and over again.

Josh Malone, a father of eight kids, understood the pain of producing water balloons in large numbers. During hot summers in Texas, his family loved to have water balloon fights but disliked the slow, hard work of making many water balloons. This problem inspired Malone to invent a brilliant, award-winning product called Bunch O Balloons to make the process quick, easy, and graceful.

While I was out buying supplies for Eva’s party, I was intrigued when I saw Bunch O Balloons at a local store. The words on the package read:

Fill & tie 100 water balloons in 60 seconds with our sensational self-tying water balloons! Say goodbye to the stress and mess of filling individual water balloons and say hello to never-ending splash out loud fun!

I was sold. I bought Bunch O Balloons for Eva’s party, and I looked forward to learning if the product could live up to its promise.

On the day of the event, I attached the stem feeding the first set of balloons to an outdoor faucet and turned on the water. In just seconds, I made more than thirty water balloons. As I pulled each perfect yellow balloon from its attached rod, a self-tying rubber O-ring contracted into place, sealing the water inside each balloon. It was so quick and easy!



I performed the same procedure for the red and blue balloons in the package and in no time had filled an entire bucket with one hundred water balloons. Just like the advertisement said, there was no stress or mess.

Eva was thrilled. I was thrilled. And during the party, the kids were thrilled as they engaged in a fun water balloon fight. Bunch O Balloons made me an excellent dad, made Eva an excellent party host, and helped her friends have a wonderful time.

Of course, making water balloons in large quantities is a relatively trivial thing—Bunch O Balloons hasn't exactly changed the world with its product. Yet, this innovative product can serve as an inspiration to anyone in any organization. What activities are you or your customers forced to perform today that are slow, hard, and ungraceful? What product or service could you invent that would give you or your customers the superpowers to make those same activities quick, easy, and graceful?

Help people be agile by removing the stress and mess of slow,

hard, ungraceful experiences.



Banking On Joy

One morning at Westpac bank in Wellington, New Zealand, a team of software developers paid a visit to the tellers who worked in their local bank branch. This was the developers' first time meeting their actual users. The visit had been arranged by Hamish King, a lean/agile coach who'd recently joined the team. He knew how important it was for developers to get to know their customers, and he decided to do something about it.

One of the developers on the team had been building software for the bank tellers for ten years. He had always worked from requirements documents, but had never once met with a real-life, flesh-and-blood teller.

Before the local bank branch opened, the tellers met with the developers to discuss problems they experienced with the software and ideas they had for improvements. Once the bank opened, the developers watched the bank tellers at work.



One customer walked up to a teller with a slip of paper that indicated she wanted to deposit cash into her “vacation savings” account. The bank teller said he didn’t know what the customer’s “vacation savings” account was—there was no such thing in his system. The transaction would require an account number.

“Why?” asked the customer. “The Westpac website allows me to nickname my accounts, and I named one of them ‘vacation savings.’ That’s where I want the money to go.”

The bank teller explained that he didn’t have access to account nicknames. “Sorry,” he said. “I’ll need your account number to proceed.”

The customer, looking visibly annoyed, managed to find the account number and completed the transaction.

When the bank closed for the day, the developers and bank tellers met again to produce a set of Post-it Notes that they could stick on the wall, each one describing a proposed improvement. To prioritize this work, they roughly

quantified the value of each improvement by looking at time saved per customer interaction, improved customer experience, reduction of errors, amount of work required to make the improvement, and so on.

After assigning values to each Post-it, fixing the issue with bank tellers not being able to see nicknamed accounts rose to the top of the list.

A few days later, the developers returned with some paper prototypes that illustrated design choices for displaying nicknamed accounts on the tellers' already-busy screens. The tellers discussed pros and cons of each design with the developers and then selected their favorite.

Two weeks later, the developers returned to demonstrate the new design that was now in production and ready for the bank tellers to use. Said Hamish King, "The joy on the faces of the bank tellers and developers was amazing to see because it was 'theirs.' Collectively, that group had worked it all the way through from ideation to production."

The collaboration between the developers and the staff of that local bank branch continued as they took on other problems and worked out improvements. Over time, the entire staff of the local bank branch adopted more agile practices, such as agile planning, retrospectives on bank operations, and voting for improvements. All of this led to better service to Westpac customers and happier bank staff.

How often do you or your teams spend time with actual users or customers? What could improve this collaboration?

Bring together people who can identify and solve problems.

Close, frequent collaboration increases agility.



Eva Learns to Ride

When my youngest daughter, Eva, was ready to learn to ride a bicycle, I started seeing young kids at the park riding “balance bikes”—small, low-to-the-ground bicycles without pedals. The kids would ride these bikes by pushing off with their feet, lifting their legs up to glide, and putting their feet on the ground to stop. Even though I learned to ride with training wheels, and Eva’s two older sisters also learned that way, I wondered if Eva could learn more rapidly using this balance bike technique.

We already had a small bike that could function as a balance bike. I adjusted the seat to the lowest position so Eva’s feet could easily touch the ground. Since I lacked the tools to remove the pedals, I figured Eva could just ignore them. With the bike ready to go, Eva grabbed a helmet and we headed for the park.

The park featured a perfect patch of flat concrete as well as a long, gently sloping concrete hill. On the flat patch of concrete, I instructed Eva to sit down on the bike, push off with her feet, and then lift up her legs to glide along. She tried and immediately fell off. She was uninjured, since she’d moved only about two feet and was initially positioned quite low to the ground. She got back up to try again, this time without any issues. After several minutes of practicing on the flat concrete, we moved to the gently sloping hill. Eva glided slowly down the hill, stopping herself about every ten feet, then pushing off again. Within half an hour, she was gliding all the way down the hill without stopping until she reached the end.

At this point, another dad and his daughter showed up at the park. The daughter’s bike had training wheels, but the dad had just finished adjusting them to be higher, so they didn’t touch the ground. I gathered that his daughter had been riding her bike with training wheels for some time now and was at the critical juncture of learning to balance without getting as much help from them. She pedaled slowly and carefully, still relying on help from the training wheels for balance.

By now, Eva decided it was time to play on the swings and monkey bars. While she played, I watched the dad completely remove the training wheels from his daughter’s bike so she could progress to the next step in her education. The girl was initially afraid to ride alone—nervous that she would fall over—so her dad walked slowly beside her, holding onto the bike.



Soon, Eva hopped back on her bike and started gliding down the hill. As the dad and daughter moved slowly down the hill, Eva easily glided past them.

Several minutes later, Eva was ready to try pedaling. She'd get one foot on a pedal, push off, and make it a few feet before putting both of her feet down again. The more she tried this, the closer she came to getting it. Then Eva announced she was ready to head home—we'd been at the park for a little over two hours.

Before we left, I asked the dad if he had any interest in having his daughter try gliding with our bike—his daughter was not making much progress with the training wheels removed. "No thanks," he replied.

The next morning, Eva and I returned to the park. On the flat patch of concrete at the entrance to the park, Eva sat down on her bike, pushed off, and began pedaling. She was riding! When she pedaled back toward me and stopped, she flashed a triumphant smile.

Eva had similar success on the gentle hill. We found some orange cones and she rode her bike in circles around them. I explained how braking worked and she practiced that as she pedaled down the slope. I was amazed that after only two short trips to the park—consisting of no more than three hours in total—Eva was biking gracefully!

This experience was starkly different and far better than how my older daughters and I learned to ride. Training wheels had made it easy for us to pedal around, but they failed to teach us balance, the essence of biking. We all experienced a great deal of fear as we attempted to ride without training wheels. In comparison, the balance bike approach helped Eva safely and efficiently learn to balance right from the start. Once she mastered balance, it was easy and graceful for her to transition to pedaling.

While more and more parents today are using the balance bike technique to help their kids learn to ride, training wheels are still as popular as ever. When I see a child learning to ride with training wheels, I feel like telling the adult next to them to stop using such an outdated and inferior approach. Since these people are complete strangers to me, I don't utter a word. But it's hard to watch.

Traditions are hard to break. Many people who learned how to perform a skill years or decades ago use the same technique when teaching others. They are stuck in a traditional mindset, unaware of new, innovative approaches that help us establish balance sooner; drive out fear more effectively; and make it quicker, easier, and more graceful to learn a new skill.

Don't just blindly follow long-established approaches when learning a new skill. Explore alternatives that accelerate learning.



Consider Your Options

After years of working without fail, my bathtub was no longer able to fill up with water. A mere trickle flowed from the faucet, and it was time to call a plumber.

The first plumber I called inspected all of the faucets in my eighty-year-old home and said, “You need new pipes throughout this old house and all the way down to the street. I’ve been in this business for three decades, and I can tell you that no other solution will solve the problem.” I asked how much it would cost. “Around \$9,000,” he said.

I called a second plumber, and after looking over my pipes, he had better news. “You only need a new pipe from the downstairs water heater to the upstairs bathroom,” he told me. “I can do that work, but it will involve breaking down part of the bathroom wall and tearing up your outside deck. I don’t put decks back together—you’ll have to handle that on your own—but I can get the new pipe installed for \$3,000.”

Well, that was much cheaper than the \$9,000 the first plumber quoted me, but it still sounded expensive and awkward—especially the deck part. I wanted to find out if there were any other options, so I called yet another plumber. This third plumber was different from the rest. He specialized in repairs to old homes and had developed an amazing solution for clogged pipes: he used a focused jet of CO₂ gas to remove blockages in old pipes and restore flow. The price to do all this? \$125. I immediately wrote a check, he performed the service, and full water flow was restored to my bathtub. The problem did not resurface, even years later.

I had nearly chosen the \$3,000 solution, which would have been a hasty decision. I knew it wouldn’t take much time or effort to explore more options, and so I did. I was lucky to find an innovative and inexpensive plumber who solved my problem with quick, easy grace.

Yet, it wasn’t *all* luck. I was resourceful by checking my options before deciding. Virginia Satir—an innovative family therapist, author, and teacher—recommended to her students and patients that they consider three options before acting on something. She said, “To have one choice is no choice; to have two choices is a dilemma; and to have three choices offers new possibilities.”¹

I love to collaborate with others to discover bargains: high value at low cost. Bargain hunting is an opportunity to be agile by considering several approaches to solving a problem and discovering a suitable solution that is both efficient and effective. Are you regularly considering your options?

To be more agile, consider at least three options.

[1.](#) Virginia Satir, John Banmen, Jane Gerber, Maria Gomori, *The Satir Model: Family Therapy and Beyond* (Palo Alto, CA: Science and Behavior Books, 1991).



Airbnb's Shipping Culture

How long does it take a brand-new software engineer at Airbnb to safely deliver new functionality to the company's website?

In many companies, the answer would be weeks or months. The general notion is that the work of newbies isn't safe to ship to production and they need lots of training and mentoring before anyone trusts them to release to production.

But at Airbnb, new hires ship within two days of being employed.

How is that possible? Aren't managers afraid these newbies will screw things up royally, putting their own jobs on the line?

Actually, the opposite is the case. Delivering value safely, quickly, and easily is something Airbnb values so much that they've built it directly into their onboarding program. The goal is to help new hires understand that they are trusted to ship to production. Here's how it works:

A mentor is paired with a new hire, and the mentor finds a defect to fix or some small feature to build. On the new hire's first day of work, the mentor helps them configure their computer (this is automated, so it doesn't take a lot of work). The mentor next helps the new hire understand their task, and then the mentor gets out of the way.

This experience gives every new hire a chance to understand Airbnb's deployment pipeline and the steps necessary to get an idea safely from their fingertips to production and into the hands of real Airbnb customers. If they have any issues, their mentor is there to help.

This approach enables agility. Airbnb cares about delivering value continuously and invests in making its deployment pipeline safe, so developers are trusted to release value to customers safely, quickly, and easily.

Drive out fear to enable agility for everyone, especially new hires.



Oprah Winfrey in Texas

On April 16, 1996, Oprah Winfrey's popular TV talk show focused on "dangerous foods" and included a discussion with experts about mad cow disease (a serious problem in England at the time).¹ One expert from the National Cattlemen's Beef Association assured everyone in the studio and television audience that American beef was safe. Another expert, a fourth-generation cattle rancher turned vegetarian, argued that it was far from safe given the widespread use of animal parts—including ground-up tissues and bones of other cattle, sheep, goats, and pigs—in cattle feed.

Alarmed by what she heard, Oprah declared on air that she would never eat another burger again. Those words sent cattle prices plummeting.

Leaders of the Texas Cattlemen's Association were furious. Believing that Oprah had lost them millions in revenue, they filed a \$12 million lawsuit against the talk show host for violating the False Disparagement of Perishable Food Products Act, a Texas food libel law passed a year earlier.

In late December 1997, a judge ruled that the lawsuit was valid and shocked Oprah's attorneys by announcing that an impartial jury would hear the case in Amarillo, Texas, starting in early February 1998.

Oprah's talk show aired from Chicago, and she was bound by a contract to broadcast two hundred shows per year. She could not simply put her show on hiatus while fighting the lawsuit in Amarillo. She could settle out of court, but she was convinced she hadn't violated any law.

She decided to fight.

Suddenly, Oprah had only four weeks to move her show and staff to Amarillo. In early January, her staff flew to Texas and found the Amarillo Little Theatre. The facility lacked the electrical power needed for the show's production equipment, had inadequate seating, and a stage that was too small. Her team upgraded the electrical power, extended the stage, and improved the seating. Then they gave the show a Texan theme (renaming it *Oprah Winfrey in Texas* and giving Oprah a cowboy hat) and booked famous Texans to be guest stars.



When the trial began in February, Oprah sat in court all day and then filmed her daily talk show in the evenings. As usual, her graceful performances on the show were enjoyed by millions.

Six weeks later, the jury agreed that Oprah was well within her rights to talk about food safety on her show and voted unanimously in her favor. As she left the courtroom, Oprah declared it a victory for free speech.

Don't let an unexpected event derail you. Handle change gracefully by being quick, resourceful, and adaptable.

1. britannica.com/story/a-brief-history-of-food-libel-laws.



Investing in Agility at American Airlines

On a Friday afternoon in early February 2020, not long after the World Health Organization declared COVID-19 a global health emergency, the Trump administration suddenly mandated that all U.S. airlines ask passengers whether they'd visited China before being able to check in for their flight. Every airline had forty-eight hours to comply. At American Airlines, executives quickly realized that it would be a real struggle to meet the government's deadline.

The following Monday, I was at American Airlines co-teaching an agile workshop in a state-of-the-art training room called "The Hangar." During class, Steven Leist, VP of Customer Technology, explained how the airline complied with the new mandate over the weekend. It's a fine example of agility in the real world.

A balanced team of executives, managers, analysts, developers, testers, and others—spread across several divisions—worked together to update every single American Airlines self-service check-in kiosk and terminal to ask the China travel question. Many of them had rarely worked together. However, for this pressing business, they quickly learned what needed to be done, collaborated effectively, and finished the work in less than twelve hours.

This rapid response was possible only because of an investment that the company had previously made. A few years earlier, Maya Leibman, American's CIO, gave IT a new mantra: *to be the best at getting better*. She and her leadership team inspired all of the organization's software teams to focus on improving four key metrics, described in the book *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*:

- *Deployment frequency*: How often the organization successfully releases to production
- *Lead time for changes*: How long it takes to get a change into production
- *Mean time to recovery (MTTR)*: The average time to recover from a failure in production
- *Change failure rate*: The percentage of deployments causing a failure in production.

A stated goal for the year was to "double deployment frequency" and improve at the other metrics. Maya and her leadership team did not impose these

metrics. They were suggestions. She wrote to everyone in IT:

These metrics aren't meant to pit teams against each other or as a means to be punitive. Rather they are intended to help you see the progress you're making on your journey. They should be a tool for learning and a validation of your hard work. Don't game the system—there's no need. Instead, as you measure, ask yourself as a team: "are we delivering value faster and improving our quality?"¹

The company's airport kiosks had historically been quite challenging to change remotely. To alter what travelers saw on a kiosk, a package of native software had to be built, tested in-house, and then deployed to all kiosks. It could take weeks or sometimes months to deploy *anything*. The situation was even worse in the case of shared kiosks used by multiple airlines. Changes to those required lengthy approval processes, and outdated operating systems made it hard to support continuous delivery.

In short, changing what customers saw on kiosks was slow, expensive, awkward, and time consuming. It would be a huge issue if and when American Airlines needed to make changes rapidly. Leadership knew this and decided to do something about it.

A group within the company was assigned the task of redesigning the kiosk software so that changes could be made with much smaller lead times and far greater frequency. The work took a few months, and once it was done, it delivered value immediately. After the redesign was rolled out, updates could, in mere minutes, change the entire user experience in American Airlines or shared kiosks across the globe. Now the airline had the ability to offer last-minute deals to passengers, thereby increasing revenue, and in the case of a public health crisis, respond rapidly to new government mandates.

Investing in improving the airline's ability to rapidly deploy value had also made the organization more resilient. David D. Woods and Erik Hollnagel, pioneers in the field of resilience engineering, observed that "success belongs to organizations, groups, and individuals who are resilient in the sense that they recognize, adapt to, and absorb variations, changes, disturbances, disruptions, and surprises—especially disruptions that fall outside of the set of disturbances the system is designed to handle."²

Speaking about American's rapid response to the China travel question mandate, Jason Hobbs, one of the company's senior IT leaders, said, "I think this was one of the first 'wow' moments where we realized we had teams that could take care of business and surprise us after the deployment-frequency investments had been made."

Improve whatever makes responding to change slow, difficult, or awkward. Invest in agility.

- [1.](#) Private email, quoted here with permission from American Airlines.
- [2.](#) Erik Hollnagel, David D. Woods, and Nancy Leveson, eds., *Resilience Engineering: Concepts and Precepts* (Boca Raton, FL: CRC Press, 2006).



A Bar, a Blender, and a Cheesesteak

Whenever I think about agile entrepreneurs, I remember the unforgettable stories in Jonah Berger's excellent book *Contagious: Why Things Catch On*. I have three favorite stories from his book. The first is about a bar, the second is about a blender, and the third is about a cheesesteak.

Two entrepreneurs once opened a hot-dog restaurant in NYC. The restaurant succeeded, and several years later they wanted to open a bar next door. How could yet another bar compete with hundreds of other bars in NYC? Put a big neon sign outside? Establish a happy hour? Conduct an advertising blitz? The entrepreneurs did something unexpected—they sprang upon the idea of making their bar a secret. Dubbed Please Don't Tell, the bar could only be entered via a telephone booth in the hot-dog restaurant. After dialing the correct number on the telephone, a secret passage opened into the bar. Servers and bartenders inside the bar asked everyone who entered not to tell anyone else about the secret bar. Of course, they did. And soon enough, there were giant lines of people waiting to experience Please Don't Tell. (I was one of them and thoroughly enjoyed the experience.)

The next story is about a blender. An engineer in Utah built an industrial-strength blender after getting tired of buying blenders that quickly broke. His new blender was powerful and resilient, yet it wasn't selling because he didn't know how to market his brilliant product. So he invited a friend to be his marketing director. One day, the marketing director showed up at the office and found the engineer blending pieces of wood. He learned that the engineer often tested his product to see what might break it. The marketing director immediately ran out to buy a video camera and a lab coat for the engineer. When he returned, he filmed the engineer, dressed in the lab coat, blending the pieces of wood. That video was the beginning of a series of remarkable videos posted by the marketing director to YouTube, under the provocative title, *Will It Blend?*

I decided to take a look—watching the engineer blend an iPhone. When he was done, all that was left was a pile of dark gray dust and a twisted metal antenna. It was pretty entertaining, so I watched more of the company's videos where marbles, golf balls, car key fobs, and other items were obliterated. Not surprisingly, the company's videos went viral, and the manufacturer, Blendtec, realized a 700 percent increase in retail sales.

The next story involves Philly cheesesteak. A person who was in charge of

all food for the W Hotels was ready to do something new. He wanted to open a luxury boutique steak house in Philadelphia. But there were already dozens of steak houses there. How could his restaurant stand out from the crowd? He decided to make a \$100 Philly cheesesteak. It was composed of the finest ingredients, including Taleggio cheese and thinly sliced Kobe beef. People in Philadelphia were outraged that a restaurant would charge so much for the city's humble and notable sandwich. Others *had* to try it. TV comedian David Letterman decided to eat one on the air. The amount of free publicity and marketing generated by the \$100 Philly cheesesteak was staggering. Suddenly, everyone was talking about the new steak house in Philadelphia, which became an overnight success.

I find all three of these stories incredibly inspiring. While most startups simply hope to survive, all three of these startups thrived. While most businesses get little attention and few new customers due to mediocre marketing, these startups got known and gained customers quickly and easily thanks to outstanding marketing. Each of these startups bucked the status quo, did something decidedly different than their competitors, and stood out from the pack. All three stories exemplify entrepreneurial agility.

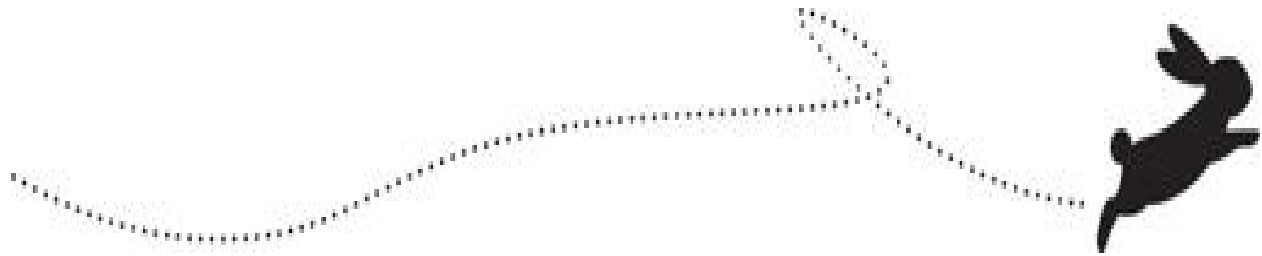
Such agility in business is uncommon. I've asked many people who don't know the \$100 Philly cheesesteak story what they would do to stand out from the crowd if they opened their own steak house in Philadelphia. The answers are almost always mundane and unremarkable.

Why is that?

Because most of us don't know the brilliant approaches that savvy marketers and talented entrepreneurs use to succeed sooner. In *Contagious*, Jonah Berger demystifies this art. He explains what "social currency" is and how it yields fantastic "word of mouth" marketing, the importance of exposing a product's "inner remarkability," and how leveraging "scarcity and exclusivity" is a quick and easy way to make customers feel like insiders. Armed with such techniques, the "sheer brilliance" in the bar, the blender, and the cheesesteak stories begins to feel more attainable.

Agility is like that. It's a skill, and this book aims to demystify it. You can learn and leverage agility to solve problems, succeed sooner, and experience the joy of agility.

Agility isn't a mystery. It's a learnable skill.



Be Quick—But Don't Hurry

Quickness is an important aspect of agility, while hurrying, rushing, or moving so fast that you are out of control is not. The mantra Be Quick—But Don't Hurry is all about understanding the difference.

While being agile requires being quick, quickness alone does not make one agile. Agility requires moving at pace, with ease and grace, or being at once quick, resourceful, and adaptable.

When we are quick, we don't hesitate; we are in control, balanced, flowing, and unhurried. We avoid taking dangerous risks.

When we hurry or rush, we are unbalanced and unstable, prone to making mistakes that may be harmful or costly to ourselves or others.

The idea is that we're focused on getting results sooner. When we accomplish things sooner, they get done faster, which is good. When we hurry or rush, we may succeed a few times, but generally it leads to problems. Some people I know are so scarred by bosses pressuring them to rush that they are permanently allergic to any notion of speed. Yet speed isn't the problem. It's rushing and hurrying, moving in an uncontrolled, reckless manner that leads to missteps.



The Wooden Way

From 1964 to 1975, the UCLA Bruins ruled college basketball, winning the National Collegiate Athletic Association (NCAA) championship ten times in twelve years. What made them so good? While the players—including Kareem Abdul-Jabbar and Bill Walton—had something to do with it, much of the credit rightfully goes to a legendary coach by the name of John Wooden.

Wooden is considered by many to be the greatest basketball coach of all time. Some have even called him the best sports coach of the twentieth century. Wooden was an exceptional basketball player at Purdue and then in the NBA, where as a player he was inducted into the Basketball Hall of Fame in 1960 before becoming a coach. As a coach, he earned a reputation for turning top players into habitual champions, and he was again inducted into the Hall of Fame in 1973—this time as a coach.

Andy Hill was one of the athletes Wooden turned into a top player. Though he had more bench time than he preferred, Hill helped the Bruins win three NCAA championships. Decades after his college days, Andy teamed up with Coach Wooden to document the coach's wisdom in a book called *Be Quick—But Don't Hurry: Finding Success in the Teachings of a Lifetime*.

According to Hill, every Bruin player heard Coach Wooden's mantra "Be quick—but don't hurry" repeated over and over, day after day. It was central to the Bruins' success, and it's a mantra that aspiring agilists would do well to understand.

Wooden explained one aspect of his mantra:

*If you hurry, you're more likely to make mistakes; but if you're not quick, you won't get things done. It's like a surgeon who comes upon things that are unexpected. If he isn't capable of responding quickly, he may lose a patient. If an attorney isn't a quick thinker, he could lose a case.*¹

Sounds like being quick, resourceful, and adaptable in the face of unexpected events. Textbook agility.

Quickness without control doesn't produce champions. Wooden knew that hurrying leads to mistakes, and on a basketball court, mistakes lose games. Opponents of the Bruins often lost because they hurried, took too many risks, and made too many costly mistakes.

Every new Bruins player had been a high school star, faster and more talented than other players. Most were prone to hurrying. Andy Hill described such players and Wooden's efforts to slow them down:

They all thought themselves capable of making any play, converting

*any shot, and stealing any pass. It was just not in their nature to ever think of slowing down; they all wanted to go faster and faster, which was why the job of slowing them down was such a priority for Coach. He devoted more teaching to this one point than to any other.*²

Slowing down was essential to what Wooden called “quickness under control.” Being in control required balance, and to Wooden, balance was everything. On defense, he insisted that his players remain in balance via excellent posture and keeping their hands close to their bodies. On offense, Wooden rejected any play that lacked balance, like fallaway jump shots, leaping-leaners, and wild drives. Andy recalled, “Over and over, in practice after practice, Coach would blow his whistle and chastise a player for being out of balance.” And Wooden didn’t just want physical balance. He said:

*If you don’t have physical balance, you cannot be quick. To have physical balance, it must be preceded by mental balance and emotional balance. If you don’t have those, you will be hurrying. Then you will have activity without achievement.*³

Losing was “activity without achievement.” It was the result of hurrying. Winning required playing quickly while under control. To do that, Wooden’s players needed to develop balance in body and brain. That meant slowing down enough to gain control. It did not mean hesitating, since hesitation impedes quickness. Andy Hill said, “Most people are naturally hesitant, and the Wooden approach was to remove all hesitation from the game.”

Mastering the mantra “Be quick—but don’t hurry” is hard. Wooden wanted his players to “operate at that edge just before it was no longer possible to be in balance, properly assess their options, and react appropriately.” That required a great deal of practice. Coach Wooden once said, “I was always more of a practice coach than a game coach—a player who practices well plays well.” Bill Walton, one of those players who went on to be in both the College and Professional Basketball Hall of Fame, described Wooden’s relentless focus on speed in practice:

*In fact, games actually seemed like they happened in a slower gear because of the pace at which we practiced. We’d run a play perfectly in scrimmage and Coach would say, “OK, fine. Now re-set. Do it again, faster.” We’d do it again. Faster. And again. Faster. And again.*⁴

Coach Wooden’s genius was in helping his players master his mantra via physical, mental, and emotional balance, slowing down enough to be in control, properly assessing options, eliminating hesitation, adapting rapidly, and being so quick that it led opponents to hurry. His formula for producing agile athletes and

winning basketball championships applies to every profession.

Stop hurrying, find balance, and learn quickness under control.

Be quick—but don't hurry.

[1.](#) Andrew Hill and John Wooden, *Be Quick—But Don't Hurry: Finding Success in the Teachings of a Lifetime* (New York: Simon & Schuster, 2001).

[2.](#) Ibid.

[3.](#) Ibid.

[4.](#) John Wooden with Steve Jamison, *Wooden: A Lifetime of Observations and Reflections On and Off the Court* (Chicago: Contemporary Books, 1997), viii.



Take the Easy Way, It Gets Harder; Take the Hard Way, It Gets Easier

Hewlett-Packard (HP), a company that is well known for its computer printers, once created the first printer in what would become a new product line of printers. This new printer included two million lines of “driver” software, the code required to turn a computer’s digital output into a printed document.

Sometime after that, HP designed the second printer in the product line, and it needed its own driver code. The designers had a choice: they could either adapt the original two million lines of driver code to work with both the first *and* second printer, or they could simply duplicate the original two million lines of code and customize it—making whatever tweaks were necessary for the second printer to work.

HP chose the latter approach. Why? Because it was faster and easier, and therefore cost less to produce. Now, however, they had four million lines of code to maintain for their two printers.

But that was just the beginning.

Over the years that followed, HP released sixteen new printers in the series, and the designers took the same approach each time. Those original two million lines of code kept getting duplicated, modified, and shipped. Eventually, a large team of programmers was assigned to maintain eighteen slightly different copies of the driver code, totaling thirty-six million lines of code!

Maintenance problems multiplied. Whenever defects were discovered, the programmers had to carefully sort through the eighteen different codebases to fix them. It was a slow and difficult process to get bug fixes out to paying customers, who likely wondered if they shouldn’t find a different brand of printer. It was also a tremendous strain on the programmers. In the software business, we call this a “maintenance nightmare.”

Rather than being resourceful and adapting the two million lines of code to work with *all* of HP’s printers in the product line, the company didn’t deviate from its “hurry-up” strategy—making the same expeditious mistake over and over again.

Jim Highsmith, a management consultant, author, and expert in business agility, noted that HP was “spending 95% of their resources just keeping up with day-to-day changes, with little left for investing in the future.”¹

Choosing the path of least resistance led HP’s printer business to become slow and ungraceful. In hindsight, management at HP referred to the thirty-six

million lines of largely duplicated code as a “black eye” that hurt both its sales and reputation in the printer market.

HP triumphed years later with its LaserJet FutureSmart Firmware program. Instead of allowing solo product managers to impose plans for individual printers, program managers helped product managers prioritize the needs of the entire printer product line, including investing time to design software that was quickly and easily adaptable.

HP had learned the lesson—“Take the easy way, it gets harder; take the hard way, it gets easier.” When quick and easy becomes slow and difficult, chances are you need to be more resourceful, adaptable, and graceful.

Don’t mistake being hasty for being agile. Take the hard way.

[¹](#). Gary Gruver, Mike Young, and Pat Fulghum, *A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet FutureSmart Firmware* (Boston: Addison-Wesley Professional, 2012).



Slack, a Prescription for Increased Agility

Tom DeMarco, one of the wisest management consultants I know and a person who has advised some of the most successful companies on the planet, once observed:

*Very successful companies have never struck me as particularly busy; in fact, they are, as a group, rather laid-back. Energy is evident in the workplace, but it's not the energy tinged with fear that comes from being slightly behind on everything. The companies I have come to admire most show little obvious sense of hurry.*¹

Just as John Wooden did, DeMarco has spent decades coaching people to avoid hurrying. In his classic book *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*, DeMarco explains why, in the name of hyper-efficiency, companies overload people with so much work that they are in a constant state of hurrying, unable to slow down, reflect, experiment, and respond to change. Without slack, defined by DeMarco as “the degree of freedom required to effect change,” an organization cannot improve or innovate, which is a recipe for stagnation.

To be clear, DeMarco isn't arguing against efficiency. He's arguing against focusing on efficiency to the exclusion of everything else. When that happens, you just get more efficient at what you already do, without making time for other vital concerns, like adapting to changing customer needs, responding to disruptive technologies, adjusting to a changing business climate, and evaluating serious competitive threats. DeMarco said:

*The overstressed organization is so busy making itself efficient that it has clean forgotten how to be effective. The two are not the same. You're efficient when you do something with minimal waste. You're effective when you're doing the right thing.*²

A “Hurry Up” organization, as DeMarco calls it, pays little attention to effectiveness and instead puts people under constant pressure to be efficient. Management regularly sets impossible-to-meet deadlines, adds extra work, expects lots of overtime, rails against any perceived waste of time, and praises heroics. Busyness rules the day.



DeMarco's message is simple and crucial: without slack, an organization's ability to be agile—to be resourceful and adapt gracefully to change—simply isn't possible. To be effective on the basketball court, Coach Wooden asked his players to slow down. To be effective in business, DeMarco asks organizations to increase slack.

So, how do you do that? First, stop overscheduling yourself and others, stop trying to do too many things at once, and stop letting efficiency dominate everything. Next, be strategic in deciding what is critical to do, and merciless in deciding what not to do. Finally, make time to reflect, improve, learn, and

change. “Improving daily work is even more important than doing daily work,” advised Gene Kim, author of *The Phoenix Project*.

By developing a capacity for change, you’ll have a greater chance of being more efficient *and* effective.

To be more agile, increase slack.

[1.](#) Tom DeMarco, *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency* (New York: Broadway Books, 2002).

[2.](#) Ibid.



Fine Products, Crafted Efficiently

As a young man, Bob Yapp wanted to learn how to design and build fine furniture. He learned about the work of a master craftsman in Europe and begged his way into an apprenticeship. Six months into the apprenticeship, the master craftsman asked each apprentice to design and build a fine wooden tool chest on their own time and present it to him as a gift when complete. No further guidance was given about what exactly to create or how to go about the work.

Bob worked tirelessly on his tool chest. He used the best materials and applied everything the master craftsman had taught him. He wanted his tool chest to be the finest one the master craftsman had ever seen!

Two years later, Bob was so pleased with his work on the tool chest that he invited his parents to Europe to see him present his gift to the master craftsman. After the unveiling of the gift, the master craftsman's eyes widened as he inspected the fine craftsmanship. After a few moments of silence, he said, "Bob, this is without a doubt the finest tool chest any student of mine has ever presented to me."



Bob turned to his parents, who were both beaming with pride. Then, just as Bob turned back to look at his teacher, he watched in horror as the master craftsman took a large hammer and destroyed the tool chest.

Bob stared at the shattered remains of his tool chest. After several moments of silence, he managed to utter one word: “Why?”

The master craftsman explained: “I gave you this assignment nearly two years ago, and while it was indeed the finest tool chest I have ever seen, a true craftsman produces quality results quickly. The tool chest you built demonstrates you are learning to design and build fine furniture, but that is not good enough. You must also learn how to build it efficiently if you want to be a true craftsman.”

Bob needed to become both efficient and effective.

Several decades later, Bob is himself a master craftsman, host of the TV show *About Your House*, and an instructor in his Belvedere School for Hands-On

Preservation. He has had his own apprentices for thirty-five years, and each one of them hears the story about his tool chest. While he's never destroyed their work, he believes that the story is highly useful. "Too often," Bob says, "people think they are masters just because they can create something nice. A true craftsperson/artisan/master is someone who creates the finest work efficiently."

Mastery includes the skill of speed. Become efficient and effective at your craft.



Stop Starting, Start Finishing

One of my company's clients is in a constant state of hurrying as it works on fourteen different initiatives with only thirty people. They have far too much WIP (work in process), which makes them highly inefficient. We've advised them to stop starting new initiatives before they finish older ones, and they completely agree! But the executives at the top of their organization keep giving them more new work to start.

What coach John Wooden called "activity without achievement" often happens when you (or your team or organization) work on too many things at once. Busyness and hurrying increase, while finishing decreases, sometimes sharply. Without focus, you won't be agile.

Sir Jony Ive, the legendary Apple designer, was once asked what life lessons he learned from Steve Jobs when he worked with him. He replied:

Steve was the most remarkably focused person I've ever met in my life. And the thing with focus is it's not sorta like this thing you aspire to or you decide on a Monday, "You know what, I'm gonna be focused." It is an every minute, "why are we talking about this? This is what we're working on." You can achieve so much when you truly focus. And one of the things that Steve would ask (because I think he was concerned that I wasn't) was "how many things have you said no to?" And I would have these sacrificial things because I wanted to be very honest about it. And so I said, "I said no to this and no to that." But he knew that I wasn't vaguely interested in doing those things anyway. So there was no real sacrifice. What focus means is saying no to something that you, with every bone in your body, think is a phenomenal idea and you wake up thinking about it but you say no to it because you are focusing on something else.¹

My friend Lynn Langit keeps a folder on her computer desktop with screenshots of things she's said no to. She finds it to be cathartic.

Agility requires focus. Say "no" more often. Stop starting, start finishing.

¹. Vanity Fair, "Apple's Jony Ive Full Conversation with Graydon Carter," Vanity Fair YouTube, October 16, 2014, www.youtube.com/watch?v=ef69BUlge-A.



Align to Accelerate

Some would argue that a team isn't even a team if it lacks alignment on a shared purpose. One of the best ways to create such alignment is with *chartering*. I learned it from my friend who goes by the name III (pronounced "three"). He explained that "chartering helps establish expectations, aligns the perspectives of all involved players, and secures commitment for the successful completion of important work."

It is not about producing an unchangeable road map. The "ing" in "chartering" means consistently revisiting and adapting the team's charter. It's also not about producing one giant document. Teams that perform chartering produce a few carefully written pages (or posters) of content. Finally, charters aren't meant to be "handed down from on high" but rather collaboratively created by a whole team.

In 2006, before my colleagues and I started building our most ambitious product (an eLearning offering), we began by chartering our work. It took us several meetings spread across several days to draft our vision (our desired future state), our mission (how we would achieve our vision), objectives (how we'd measure the success of our work), our community (who would be involved in the work), and our working agreements (how we would work together harmoniously). This work allowed us to get aligned and focused on a shared purpose.

Over the succeeding months and years, we refined and improved our charter. Whenever choices came up about what to do or not do, we consulted our charter. It informed our actions and decisions, helped us stay focused on our vision (which was "Rapidly spawned, self-taught developers, skilled in the arts of extreme programming"), and enabled us to accelerate toward a shared purpose.

A lesson I've learned frequently is that slowing down to create alignment within a team, before rushing to begin work, helps the team accelerate. Without alignment on a shared purpose, a team can waste time doing work that doesn't add value or fit a desired outcome. Misalignment with sponsors, within a team, or across teams is all too common.



One could easily write an entire book about chartering. In fact, two friends of mine, Diana Larsen and Ainsley Nies, have done just that. Their excellent book is called *Liftoff: Start and Sustain Successful Agile Teams*.

Alignment is key to acceleration. Avoid rushing into work without it.



Maximize the Amount of Work Not Done

Alexandre Martins worked in the software division of Globo, Brazil's largest television production company and the fourth largest broadcaster in the world. One day, he and his team were asked to rewrite an old but important piece of software that managed user identification and authorization. The service was used by many of Globo's systems, but it was old and needed to be modernized.

When rewriting old software, many developers simply assume they must rewrite *everything*. They believe that the new software should do exactly what the old software did, only using new technology.

Alexandre had a different idea. As an advanced technologist and serious student of lean and agile thinking, Alexandre wanted to be both efficient and effective. Rather than rushing into a rewrite, Alexandre began by figuring out what work was important to do and what work could be eliminated.

He and his team used a tool to measure the usage of features in the old software. It didn't take long to discover that a whopping 40 percent of the features were *never* used! That meant that he and his team only needed to rewrite 60 percent of the old software, a massive savings in time and effort.

Simply by measuring usage first, 40 percent of work was eliminated, enabling Alexandre and his team to reach a better outcome sooner with far less effort. That is quickness under control, efficiency, *and* effectiveness and a wonderful example of agility.

To increase agility, do less. Find the right work to do, and
maximize the amount of work *not* done!



Fast, Frugal Feedback from a Feature Fake

At a conference in San Francisco, Brad Smith, the CEO of Intuit, once told a story that inspired me. He explained how users of TurboTax, the company's best-selling software, were able to help each other via the product's Live Chat feature. Given a question about something on a tax form, a TurboTax user could submit the question to TurboTax support and wait for a response, or they could immediately engage with other users and possibly get their question answered right away.

After the conference, I mentioned this story to my work colleague John Tangney, who quickly responded, "Yes, I love that feature! I used it the other day and someone in the chat helped me with a tax question."

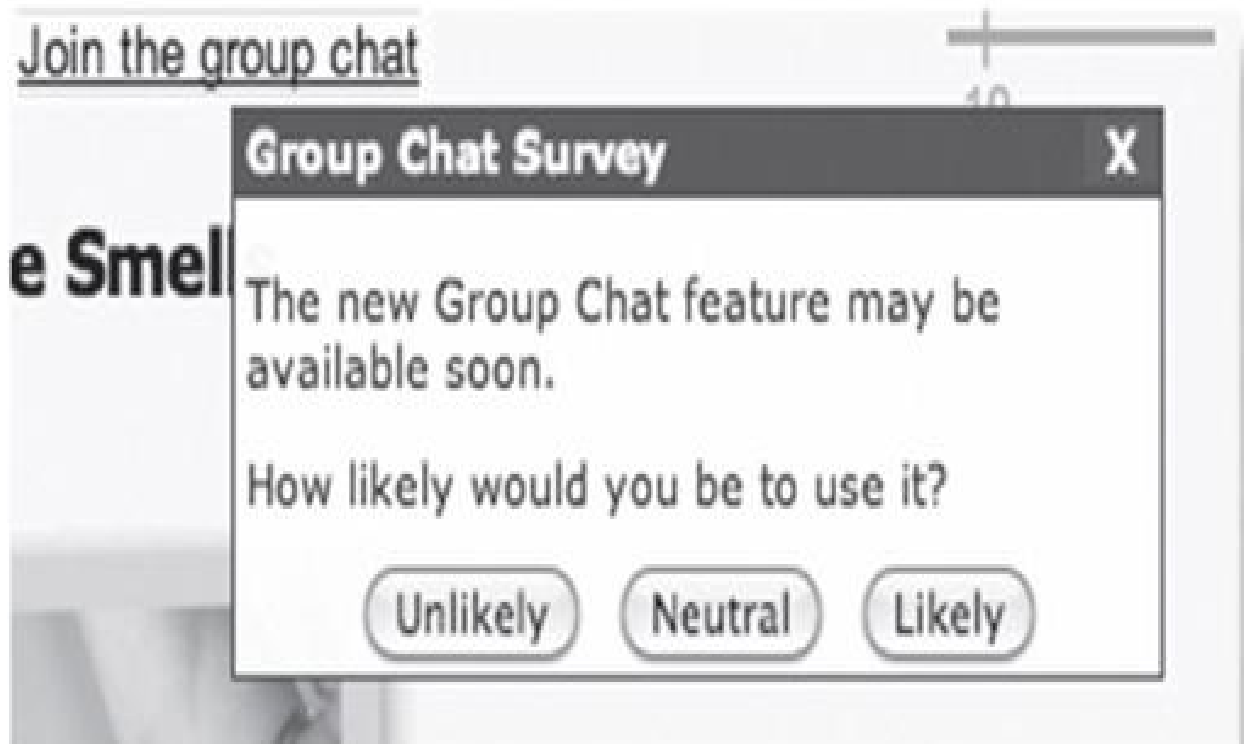
Brad's and John's stories got me thinking about users of my company's eLearning software. Maybe our students needed a similar Live Chat feature so they could spontaneously discuss ideas or answer each other's questions. But rather than rush to build this feature, I first decided to quickly test whether students even wanted it.

I used a fast, frugal user-research technique called a feature fake, a quickly fabricated fake version of a feature that enables a software team to rapidly learn whether or not to even build the feature.

Within three hours, John and I had developed and deployed a fake group chat feature on our site, along with a report to help us measure user interest. The fake feature, which sat at the top of every eLearning page users saw, looked like this:

16 users from 4 countries are online right now. [Join the group chat](#)

When a user clicked "Join the group chat," they would see this little survey pop-up:



Users could close the pop-up or choose to answer our survey. Here are the statistics we gathered after one hundred unique users saw the fake:

Unique logins with feature present	Users who tried the feature	Users who took survey	Users who clicked "Unlikely"	Users who clicked "Neutral"	Users who clicked "Likely"
100	22	14	1	7	6

Our fake was live for three business days, during which one hundred unique users saw it. The data did not provide compelling evidence that we ought to build the real group chat feature. And while we could have attempted more tests, using slightly different language for the group chat feature fake, we doubted that the results would be substantially different.

In many professions, teams often hurry to produce speculative features or services that are either never or seldom used by customers. It's often better to slow down and first explore whether or not a new thing really needs to be produced. I was delighted with the speed at which we learned not to produce the group chat feature. A quick and easy experiment helped us become more agile,

saving precious time and money.

Get fast, frugal feedback on what does or doesn't add value.

Before adding more, explore!

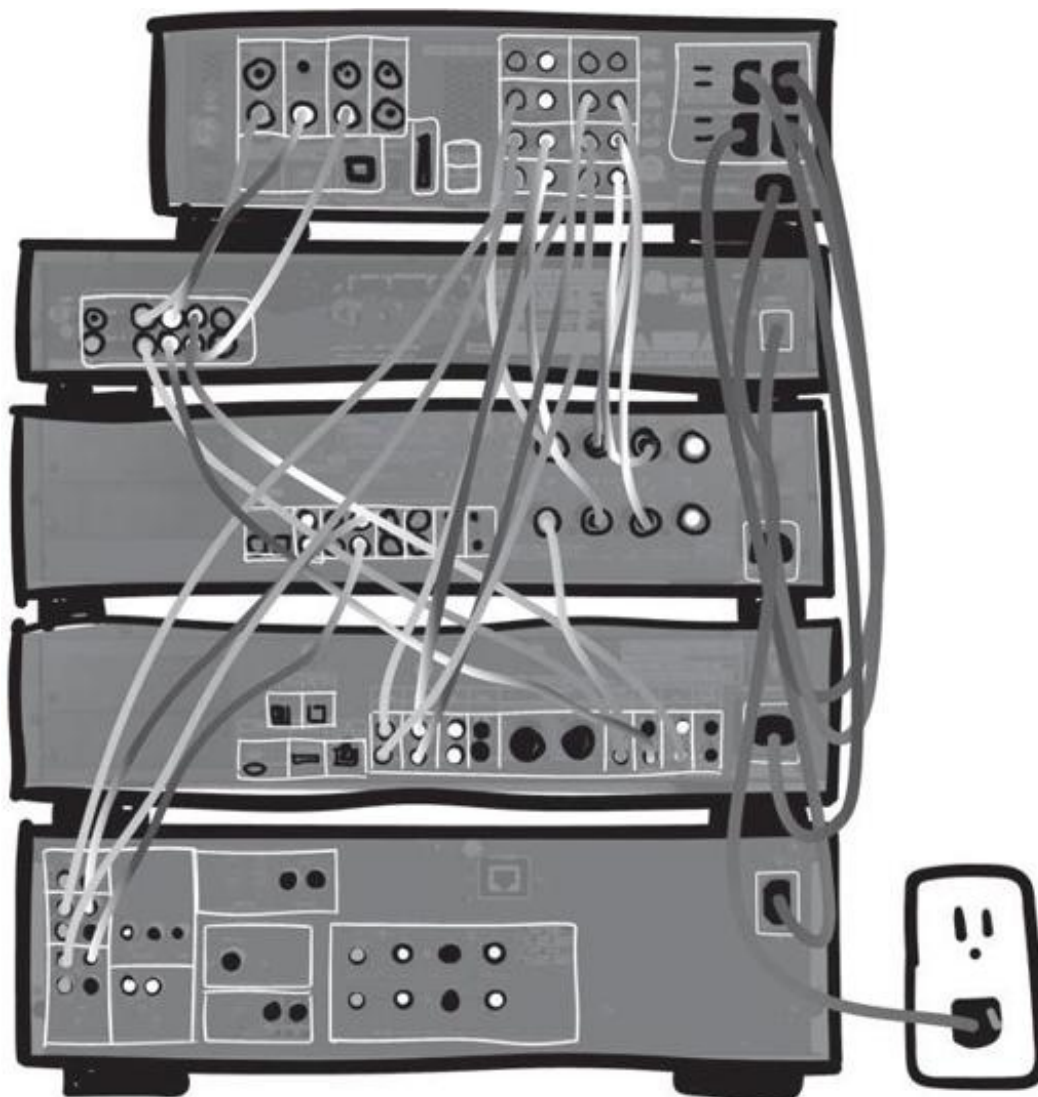


Speed Up by Taking Smaller, Safer Steps

Decades ago, people often had a large stack of electronic components to play music and watch videos. Typical stacks included a cassette tape player, a compact disc (CD) player, a VCR or DVD player, a television, a cable box to receive and decode the television channels, and maybe even a turntable for listening to vinyl records.

But that wasn't all. You would also need an audio/video (AV) receiver to take all of the audio and video sources from the devices, amplify them, and send sound to your speakers and video to your TV. Typical AV receivers enabled you to tune into AM and FM radio stations, adjust the volume, control bass and treble levels, and act as a master power switch for every device.

The assembled system was quite impressive—when it worked—but integrating all of these devices to work correctly was never an easy task. You'd have a bunch of power cords, AV cables of different kinds, and numerous input and output jacks on the back of every device. Which cord was the correct cord to use, and which input and output jack was it supposed to be plugged in to?



Years ago, when I moved from New York City to an apartment in San Francisco, I went through the painful process of reconnecting all of my devices. Once they were stacked, I attached all of the cords and cables, hoping I'd selected the correct input and output jacks. Mistakes were common. Why wasn't sound coming from the speakers? Why wasn't the TV getting a signal from the VCR? It took several maddening hours of my time to sort out the mess.

But that changed when a few years later I moved to a house in Berkeley. I still had all of the same audio and video devices to deal with, but *I* had changed in the interim. In my software development work, I had learned the value of working in small, safe steps, testing all along the way. Now, standing in front of my stack of unintegrated audio and video devices in my new home in Berkeley, I wondered whether this approach could help with *my* integration problem.

I began with the AV receiver. Since it wasn't plugged in, it couldn't play the radio or perform any functions. I had a failing test. I plugged it in, attached the

antenna and speakers, and confirmed that the radio worked. Then I adjusted the antenna and speaker cables so they were neat and tidy. Next, I focused on the cassette tape player. It also didn't work, so I plugged it in and attached its power cable and cords to the receiver, then tested whether it worked. It did not. I had selected the wrong input jack. After correcting my mistake, I validated that all functions on the cassette tape player worked and that the receiver controlled it correctly, including acting as the master switch for its power. I adjusted the cable and cords to be nice and tidy.

I repeated this process with the remaining components, one after another. To my delight, within about fifteen minutes, everything worked! By not trying to hurry my way through the process, I saved time.

Taking smaller, safer steps and validating continuously is a practice I'd learned from Extreme Programming, and it helped make a normally painful and error-prone hardware integration process quick, easy, and graceful.

That was a revelation to me. Over time, I've come to see that smart practices in one field often have broad applicability across other fields.

To go faster, take smaller, safer steps.



Size Teams for Few to No Slow Handoffs

Phil Ensor, a management consultant who invented the term *functional silo syndrome* in 1988, likened the grain silos he would pass on his drives through Illinois to the silos he found in organizations. A siloed team tends to focus on its own work and doesn't collaborate gracefully with other siloed teams. When an organization needs work to be done across silos, you end up with slow handoffs—one silo waiting a while for another silo to complete work.

Teams that are truly agile work quickly and easily and have few to no slow handoffs. They are composed of the right mix of staff (something we call a “balanced team”) to enable high-speed value creation and delivery. Two of the most agile teams I've ever worked with were not small. One team had sixteen full-time people (and several more part-time staff) while the other had thirty full-time people.

Their effectiveness goes against the literature on ideal team size, which cites “scientific” research about how the highest-performing teams are usually around four to six people. The only trouble is, most of this “ideal team size” research wasn't conducted with deeply agile, balanced teams. And none of this research discusses the slowness of handoffs between small teams.

An Amazon software team can be fed with just two pizzas and has the autonomy to deliver valuable work to customers without having to wait on slow handoffs to other teams. This concept has become enormously popular. The trouble is, many organizations that have adopted “two-pizza teams” have failed to give them autonomy to work without slow handoffs to other teams.

Having one giant team isn't the answer either. I once worked with a 180-person team at a large bank. This team was extremely slow at delivering value. Managers on the initiative competed with one another to hire and manage the most people. That's how the team mushroomed to 180 people! After several months of coaching this huge team and understanding the nature of their work, it became clear that what they were doing could be done far more efficiently with about twenty people.

If your team has slow handoffs with other teams, here are some ways to begin removing or reducing slow dependencies:

- *Remove one slow handoff at a time:* Look for one slow handoff and remove it by inviting an external person to collaborate (even temporarily) with your team. Repeat for every slow handoff.

- *Insource slow dependencies:* For any slow dependency, find a way to insource it completely rather than outsourcing it to another team.
- *Make the slowness visible:* Calculate and visualize the time wasted while waiting for slow handoffs to complete. Use metrics and visualizations to catalyze a change toward a genuine, balanced team.
- *Limit work in process:* While a team is waiting on other teams to complete work, it often begins new work. This leads to lots of work in process, with little getting finished and delivered. Limiting the amount of new work that may be started before current work is finished is a way to bring more attention to the bottlenecks resulting from slow handoffs.

If your organization suffers from the functional silo syndrome, it can be particularly challenging to create autonomous, balanced teams. In such situations, “innovation by isolation” can help. Get permission to experiment outside of the organization’s structure so newly formed teams aren’t subject to the organization’s normal bureaucracy.

Forming or evolving to autonomous, balanced teams may seem hard at first, but it’s one of the best-kept secrets for removing delays and being more agile in delivering value sooner. To create such a team, consider what full-time staff you need and what part-time staff you will need on occasion (e.g., lawyers, auditors, advisors, etc.). Help the team understand its primary goal and then watch it execute quicker, easier, and with far more grace than siloed teams that are continuously delayed by slow handoffs.

Remove or reduce slow handoffs to increase agility and escape
the functional silo syndrome.



Doing the Impossible Fifty Times a Day

One Sunday morning, during a business trip to the Netherlands, I woke up in my hotel room, checked my email, and discovered that while I was fast asleep, one of our eLearning customers had reported a defect.

I quickly investigated the issue, wrote an automated test to confirm it, found a fix, and made the change. My work kicked off a build, which ran our automated tests to check whether my change was safe. The build passed all of these tests, and my change was automatically pushed to production. Within minutes, the fix was live, and I happily emailed the individual who reported the problem to let them know it was fixed.

All before breakfast.

Continuous deployment enabled me to fix a customer's reported defect with quick, easy grace. It was quick because it only took a few minutes to release. It was easy because I didn't have to do a lot of work to get the fix into production. And it was graceful because it was effortless to release, and I didn't have to disturb my colleagues, who were still asleep back in North America.

I first learned about continuous deployment in a 2007 talk given by Timothy Fitz, a developer at a company called IMVU. His talk, called "Doing the Impossible Fifty Times a Day," explained how he and fellow IMVU developers deployed new code to production on average fifty times per day. To many, it seemed either crazy or unreal.

Back then, even sophisticated software teams that practiced Extreme Programming released software to production only every week or two. *Continuous integration*, a process of continually combining all of the programming work of many engineers throughout the day (rather than waiting days or weeks to do so), was considered state of the art. Few programmers had yet conceived of *continuous deployment*.

Timothy's talk was *revolutionary*. He explained how IMVU had gradually automated their entire build, test, and deploy process to be quick, easy, and graceful. Every time a developer committed a change, a build would kick off, thousands of high-speed, automated tests would run, and if the build passed the tests, the code would automatically begin making it onto a large cluster of computers that automatically monitored to ensure that the release was safe. If a problem was detected, the release was automatically rolled back.

Best of all, customers never even knew that the software they were using had

just been upgraded. Zero-downtime deployments were integral to continuous deployment and a far cry from seeing notices like “Our website is down for maintenance.”

Soon after hearing Timothy’s talk, I knew that we needed to try continuous deployment with our own eLearning product. At the time, we released changes to our eLearning once a week. As Extreme Programming practitioners, we already had a very disciplined and safe development process. We’d written thousands of our own high-speed, automated tests to check that our software behaved correctly. Occasionally, one of our acceptance tests would fail for no good reason.

To implement continuous deployment, we first needed to make sure that all of our tests could run thousands of times without fail. Once that was done, we needed to automate our deployment process and engineer a way to support zero-downtime deployments. Finally, we had to configure various production computers to enable our unique version of continuous deployment (which was less sophisticated than IMVU’s).

I remember the fear and excitement we all experienced once we could finally deploy continuously. It didn’t take long for us to love it. We no longer needed to coordinate around a weekly deployment. Changes made to our software throughout the day magically rolled out to production. We noticed an immediate reduction in our stress.

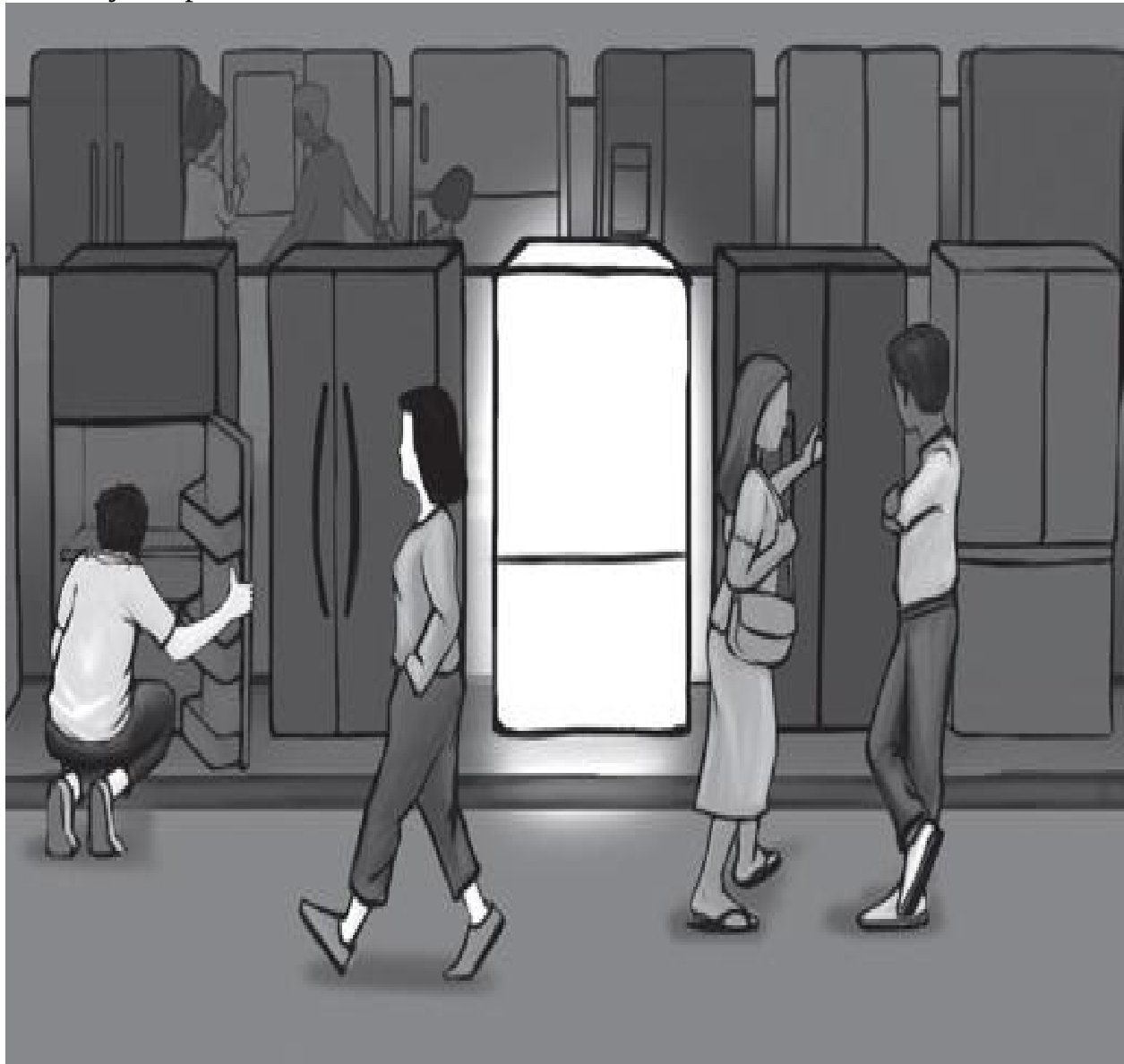
Today, more and more organizations are leveraging continuous deployment to deliver a steady stream of improvements (both new capabilities and lightning-fast defect fixes), learn rapidly from fast, cheap experiments, and adapt quickly to unforeseen events. Continuous deployment enables agility in software delivery and embodies Coach Wooden’s idea of “quickness under control.”

If something is slow, stressful, or awkward, transform it via safety
and speed.

● ● ● Haste Makes Waste

One evening during a business dinner, a longtime employee of Whirlpool, an American manufacturer of washers, dryers, and other appliances, told me the story of how the company initially sold their products in China. He explained how they studied Chinese appliances and made sure their appliances were equivalent feature for feature. When they shipped their appliances to Chinese stores, they expected to start seeing significant sales.

Only the products didn't sell!



Whirlpool sent a team to China to find out what was wrong. The team went to stores and spent time with people who were about to buy a new appliance.

They asked people why they didn't consider buying Whirlpool's products. The answer was immediate: "Your appliances are white, a color that we associate with mourning and death. We prefer colors like yellow or red, which symbolize wealth and abundance."

Armed with this new data, Whirlpool changed the color of its Chinese appliances from white to yellow and red. As soon as they made this change, Chinese customers started to buy them.

Could Whirlpool have saved time and money by experimenting and learning faster? Certainly. It could have questioned its own assumptions about color and learned sooner that its "funeral fridges" would never sell in the Chinese market. Its haste to bring the appliances to a new market resulted in a lot of waste and almost doomed the initiative.

Proactively learn what is or isn't needed to succeed sooner.



Don't Make People Wait

My company once won a large contract with the well-known Wall Street bank Morgan Stanley. One day, while visiting the bank, I asked an executive why his firm chose my company for the work. The executive replied, “We needed more help, and your competitor didn’t answer our emails, so we called you.”

That response taught me how important rapid responsiveness is to revenue. My competitor already had a few people placed at Morgan Stanley and could have easily grown its footprint at the bank, thereby significantly increasing revenue. Instead, my company got the work and the revenue.

Ever since that day, I’ve told the Morgan Stanley story to every new employee at my company. I consider rapid responsiveness to customer needs a financial safety practice. It’s neither safe nor respectful to make people wait, whether those people are potential or existing customers.

I’ve found this is particularly true when it comes to sending proposals. After meeting with a potential client to discuss their needs, we promise to send the client a proposal. How long does it take for us to deliver that proposal? I’ve found that the longer it takes, the less business we win. Customers simply don’t like to wait. This has led us to optimize every aspect of proposal writing and delivery.

In his *Harvard Business Review* article, “The Most Important Metrics You’re Not Tracking (Yet),” Gene Cornfield points out that while most leaders think their companies are customer-centric, they tend to track company-centric metrics like revenue or profit rather than the things customers actually care about. For example, Quote Turnaround Time is a customer-centric metric because it measures how long customers wait to get a quote. If you are customer-centric, you’ll constantly find ways to avoid disrespectful delays and improve customer responsiveness.

When our most important customer recently asked me for the status on our recruiting efforts to help staff a critical software initiative, I didn’t reply for three days! Why did I delay for so long? I was in a series of all-day, strategic meetings. In hindsight, I should have quickly delegated this status update to a colleague who was available to respond rapidly. Instead, three days after my customer’s inquiry, I sent a reply that began with an apology for the delay.

I later learned that because of my slow response, this customer started seeking staffing help from other companies. Fortunately, we were ultimately able to satisfy their needs, but it was a close call that could have hurt our business

with this customer—perhaps even lost it—and cost us a significant amount of revenue.

Respond rapidly to people. It respects them and rewards you.



Ward's WikiWikiWeb

In 1994, the World Wide Web was experiencing a meteoric rise thanks to Mosaic, the world's first graphical web browser. At the same time, a small community of programmers became interested in documenting patterns (common solutions to recurring problems) in software design. Ward Cunningham was one of them. After attending an inaugural conference on software patterns, Ward considered how best to support colleagues who wanted to document knowledge about patterns.

He quickly realized that the World Wide Web would be the ideal place. With some inspiration from a 1980s Apple technology called HyperCard, Ward wrote software that allowed visitors to a web page to author content on that page and easily cross-link pages. It was unusual back then to allow strangers to write *anything* on your web page. Every HTML book made it clear that, for security reasons, web pages ought to be “read only” so no one except the owner could modify them.

Ward ignored this advice. He wanted to make content authoring on the web easy and quick, requiring no knowledge of HTML. Remembering the word *wiki*—Hawaiian slang for “quick”—from riding in the airport Wiki Wiki Shuttle with his wife on their honeymoon to Hawaii, Ward decided to name his new software WikiWikiWeb, using *wiki* twice to emphasize “very quick.” On March 16, 1995, Ward published the Portland Pattern Repository, a collaborative place to share and learn about people, projects, and patterns. It was powered by the world's first wiki.

I first visited Ward's wiki in 1996, led there by my interest in software patterns. It was a welcoming place where people were trusted to contribute and collaborate on content. You didn't even need an account (i.e., a username and password). It felt revolutionary, wonderful, and *agile*. I began contributing content to the wiki and making daily visits to the Recent Changes page to learn more about what was being added and who was adding it.

I also remember thinking Ward had no sense of graphic design. The icon he used for each wiki page looked faded out and rough, as seen on the next page.

Little did I know that this was intentional. After becoming friends with Ward, I learned that he was a big fan of *wabi-sabi*, the Japanese philosophy and design aesthetic that values the imperfect, impermanent, and incomplete over the perfect, permanent, and completely formed.



The *wabi-sabi* aesthetic pervades wiki. Nothing is perfect, unchangeable, or ever complete. What emerges on wiki is unpredictable. Visitors participate in creation, collaboration, and continual refinement, helping turn early, rough content into mature, well-written ideas. People within the community also have the freedom to delete content.

In 1999, Jimmy Wales founded a free online encyclopedia called Nupedia. Designed to compete with professional encyclopedias, Jimmy and Larry Sanger—the site's editor in chief—recruited volunteer contributors with expertise in a subject and then required them to go through a seven-step approval process before publishing any content. Because of this rigorous process, a total of only twenty-one articles were published during Nupedia's first year, far too sluggish to be viable. Jimmy and Larry started looking at wikis, and on January 15, 2001, they launched Wikipedia, which enabled live, wiki-based article creation and updating. Volunteers, now unencumbered by Nupedia's slow approval process, produced more than eighteen thousand articles during Wikipedia's first year.

Wikipedia spurred explosive growth in wikis and helped make them a household name. In March 2007, the word *wiki* entered the *Oxford English*

Dictionary. And in 2010, Wikileaks brought a great deal of attention to wikis after thousands of classified U.S. government documents were leaked on the site.

By being unencumbered with excessive rules and regulations, Ward's wiki makes it easy to be quick and adaptable. Minutes after a noteworthy event occurs in the world, you will find new or updated content about it on Wikipedia. That content may evolve as more information is crowdsourced. Wikis become highly adaptable places in which ideas grow and collaborations form in unpredictable ways. It's a shining example of agility on the World Wide Web.

Avoid many rules, regulations, and restrictions. Make it easy to be quick and adaptable.



Despots and Dentures

To say that my teacher was furious would be an understatement.

I'd just entered my high school social studies class, and our normally sweet and gentle teacher was beside herself with anger! She glared at us and barked out new rules for us to follow—starting *immediately*! There would be penalties for handing in homework late, being tardy to class, and not paying attention. And there would be serious consequences for our grades if we failed to follow her new rules.

When a few students wandered into class late, she yelled at them! Our entire class was speechless, traumatized by what we were experiencing.

Then, all of a sudden, her stern expression turned into a big, warm smile. She said, “You’ve just experienced what it’s like to live under a despot.”

I will never forget those words.

What an incredible lesson! We could feel it in our bones. There would be no boring lecture about despots and despotic governments. She wanted us to experience for ourselves what despots were like and what it was like to live under one.

Our teacher was a master of accelerated learning. Her performance was agile: she delivered an unforgettable lesson with quick, easy grace.

Several years earlier, my grandfather had given my brother and me a similarly visceral lesson. The lesson was about our teeth and the importance of brushing. We were sitting at the kitchen table, and he reached into his mouth and removed his dentures. What?! Grandpa’s teeth aren’t real? We were shocked.

As our grandfather showed us his dentures, he explained, “*This* is what happens when you don’t take care of your teeth.” Again, an unforgettable lesson, taught quickly and easily. Thereafter, we’d brush regularly, without reminders from anyone.

I’m a big fan of experiential learning guru Sivasailam “Thiagi” Thiagarajan. Thiagi teaches educators how to use games, simulations, and *jolts* to quickly and easily deliver impactful lessons. That’s exactly what my teacher and grandfather did—they delivered learning jolts that have stayed with me my entire life.

Accelerate learning via eye-opening experiences or examples.



Don't Converge Prematurely

A few years ago, I was working with my colleague Tasha on a graphic for Industrial Logic's annual holiday card. Tasha showed me eight static images that she liked, and from that group we chose the two best ones.

While they were both really good, one stood out. It was a striking black-and-white image of a tree with no leaves surrounded by a field of snow. Tasha and I decided to use this image. We then worked on the fonts and holiday message for the cards, and pretty soon we were done.

Before sending the holiday card to clients, we decided to get feedback from our colleagues, including Miguel, our graphic designer in Brazil. Miguel said, "You know, it looks kind of like a dead tree in a stark environment. You think this is a good holiday graphic?"

Uh-oh! Miguel was right! We hadn't looked at the image that way, but once our colleague mentioned the dead tree, we couldn't unsee it. We also realized how a winter holiday scene was inappropriate for people in the southern hemisphere, who were experiencing summer. We ended up reworking the entire card, using a completely different image and layout.

In this case, we prematurely converged on a suitable image for the holiday card. If we had slowed down a little and sought a greater diversity of opinion earlier—when first selecting static images—we would have learned faster and completed a better holiday card sooner. Our diverse customers around the globe would have been better served.

No matter how agile you believe you are, there's always more to learn. Seek diversity of opinion early and often to produce better outcomes. Don't rush learning!



Fail Fast (But Don't Rush)

I recently bought a heavy oak bookcase and wanted to put it in a corridor on the second level of my home. My house sits on a hill and has three levels, the first of which is on top. To get this bookcase to the corridor on the second level, the delivery people would need to carry it halfway down some stairs, make a 180-degree turn, descend some more stairs, and then make a final ninety-degree turn.

The delivery people weren't sure they could do it. They asked if I had an alternate location. I did, and that was to put the bookcase on the third and lowest level, which is accessible by going around the house and doesn't involve navigating winding stairs.

But I believed the bookcase would look best in that second-story corridor, so I asked the delivery people to reconsider. They used measuring tape to analyze the situation, but it still didn't look good to attempt it.

I was about to give up when I had an idea for learning rapidly: What if we took the bookshelf out of the box it came in and then used that box to test how hard it would be to get the bookshelf to the second-story corridor? The delivery people agreed with this plan and they maneuvered the lightweight cardboard box down the stairs to the second-story corridor. It worked! But it was a tight maneuver and they explained that it was very possible they would damage the floor, walls, or ceiling while moving the bookshelf.

I decided not to risk it and had them go ahead and deliver the bookshelf to the third level of my home.

Thanks to the cardboard box, this failure was quick, easy, and graceful. Nothing was damaged and not much time was wasted. Failing fast involves experimenting quickly and cheaply, without investing lots of time, money, or energy.

I wasn't so lucky with a stand-alone office I built in the backyard of my previous home. The new office was tiny (twelve feet by ten feet) and the backyard wasn't large. The contractor asked me where I wanted to position the office. A work colleague suggested putting it on the right side of the backyard, so a large deck could be erected on the left side. I liked that idea and quickly went along with it. In hindsight, I was rushing, as I was very busy with other work.

After notifying the contractor of the desired location, concrete was poured for the foundation and the wooden walls of the small office began to take shape. I looked out of one of my daughter's bedroom windows and realized that the

new office blocked her view! It was so awful that I made the difficult and expensive decision to have the contractor relocate the office, which required pouring more concrete for a new foundation and using a hydraulic lift to change the location of the already-partly-constructed office. Ouch!

Failing fast is an agile way of learning. It's useful in many human endeavors.

If I had slowed down a bit and not rushed, I could have suggested a technique I learned from the writings of architect Christopher Alexander: erecting paper or cardboard walls to simulate where the new office could be located and evaluating the decision quickly and cheaply *before* jumping into expensive, hard-to-change construction.

This is the difference between hurrying and being quick. "Slow is smooth, and smooth is fast" is how the U.S. Navy SEALs describe it. Coach John Wooden calls it "quickness under control." It's a useful goal, whether you are playing a sport, performing a job, or just doing something around the house.

Learning rapidly is agile. Learning by rushing is not.



Summary: Be Quick—But Don't Hurry

Coach Wooden trained his UCLA Bruins to “operate at the edge just before it was no longer possible to be in balance, properly assess options, and react appropriately.” To do that, you must learn to stop hurrying, find balance, and learn “quickness under control.”

Rushing or being hasty has no place in agility. To become more agile, you often need more slack, slowing down sufficiently to make room for improving.

Mastery includes the skill of speed. Make it your goal to become both efficient and effective at your craft.

Agility requires focus. That means saying “no” more often. Stop starting new things and start finishing what you’ve already started. Teams that are aligned on purpose accelerate faster toward achievement.

To increase agility, do less. Find the right work to do, and maximize the amount of work *not* done! Get fast, frugal feedback on what does or doesn’t add value. Before adding more, explore!

To go faster, take smaller, safer steps. Remove or reduce slow handoffs to increase agility and escape the functional silo syndrome.

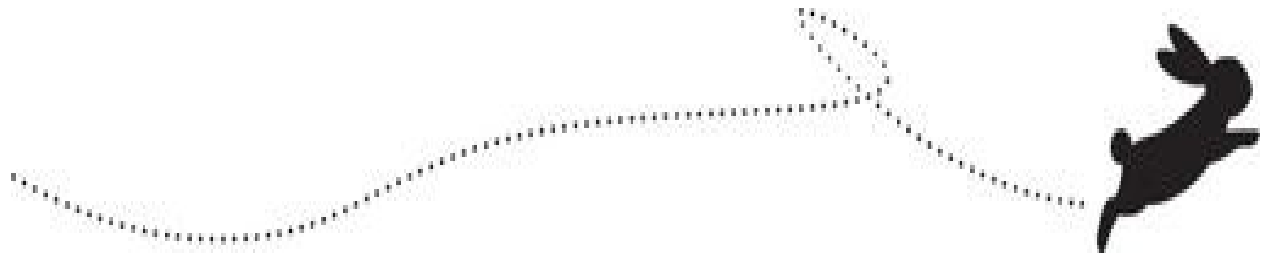
If something is slow, stressful, or awkward, transform it via safety and speed. Proactively learn what is or isn’t needed to succeed sooner.

Respond rapidly to people. It respects them and rewards you. Avoid many rules, regulations, and restrictions. Make it easy to be quick and adaptable.

Accelerate learning via eye-opening experiences or examples.

No matter how agile you believe you are, there’s always more to learn. Seek diversity of opinion early and often to produce better outcomes.

Learning rapidly is agile. Learning by rushing is not.



Be Balanced and Graceful

Agility requires balance. When we are physically, mentally, and emotionally in balance, we are stable and prepared for change. When we are unstable or out of balance, perhaps due to hurrying, rushing, or being disturbed, we frequently make mistakes or hurt ourselves.

Balance is required to be quick. Balance in business is critical for succeeding sooner. Imagine a strategic meeting of ten people, during which everyone gets an opportunity to speak and be heard. Now imagine the same meeting with only a few people dominating the conversation. The first meeting has more balance, increasing the chance of leveraging intelligent ideas from multiple minds. That's how you achieve better outcomes sooner.

A balanced board for an organization has an appropriate mix of genders and ethnicities. A balanced team has the right staff needed to succeed. A balanced portfolio has a thoughtful ratio of risk to safety. A balanced product strategy includes the right mix of production and maintenance.

Balance, in any endeavor, enables grace. When our movement is graceful, we are poised, flowing, perhaps even elegant. An athlete may execute a move in their sport gracefully; a surgeon may perform a procedure with grace; a manager may handle a tricky personnel situation gracefully; a team may commence and complete work quickly and gracefully.



Habitual Excellence

In 1987, soon after becoming Alcoa's CEO, Paul O'Neill visited the company's plant in east Tennessee. While walking around, he noticed workers sweeping sawdust onto oil spills. Thinking that odd, Paul asked where the oil was coming from. He learned that the plant's forklift trucks routinely leaked oil onto the floor and the sawdust prevented people from slipping and injuring themselves.

The job of sweeping sawdust onto oil spills had existed for several decades, and O'Neill quickly concluded that it was pure waste. Unlike cleaning staff who sanitize an operating room and are therefore essential to keeping surgical patients safe, the work of sweeping sawdust onto oil spills existed solely because of an ongoing equipment failure. But what bothered O'Neill even more was the failure to treat those workers with dignity and respect by giving them meaningful, value-adding work.



Why hadn't the supervisors identified this waste and eliminated this undignified work? Dr. Allen C. Ward, a leading authority on Toyota's product development process and the author of *Lean Product and Process Development*, observed: "The hardest part about developing eyes for waste is that most waste is caused by doing things right in the conventional system." The sawdust sweepers were removing a hazard and thereby making the plant safer. Supervisors never even conceived of eliminating work that made the plant safer.

O'Neill asked the supervisors in Tennessee why the forklift trucks leaked oil. The supervisors patiently explained that it was just something forklift trucks did. Unsatisfied with that answer, O'Neill flew to Japan to buy forklifts that didn't leak oil from heavy equipment manufacturer Komatsu. When those new forklifts made it to Alcoa, the decades-old job of sweeping sawdust onto oil spills was eliminated and the workers who performed that job were reassigned to more meaningful work.

While he is best known for safety leadership, Mr. O'Neill's higher mission was something he called *habitual excellence*. He believed that in order for any organization to have a *potential* for habitual excellence, everyone in the organization needed to answer yes to the following three questions:

1. Are you treated with dignity and respect every day by everyone you encounter without regard to race, or gender, or nationality, or pay level, or rank, or any other qualifying characteristic?

2. Can you say, "I'm given the things I need, education, and training, and financial support, and encouragement—that's really important—so that I can make a contribution to this organization that gives meaning to my life"?

3. Are you recognized every day by someone whose judgment you value?

I've anonymously polled staff in my own company on O'Neill's three questions. The results revealed areas for improvement, such as providing more training and more frequently acknowledging people's contributions. The sawdust sweepers story helped me grasp the depth of O'Neill's first question. Whereas I had assumed that treating people with dignity and respect mostly involved social interactions, now I understand that it also means caring about the very nature of their work.

Habitual excellence isn't possible if management doesn't care deeply about people. While many leaders work to increase efficiency, improve profits, grow market share, or improve quality, great leaders like Paul O'Neill focus instead on helping people thrive at work.

To achieve habitual excellence, care deeply for people.



Mixed Messages

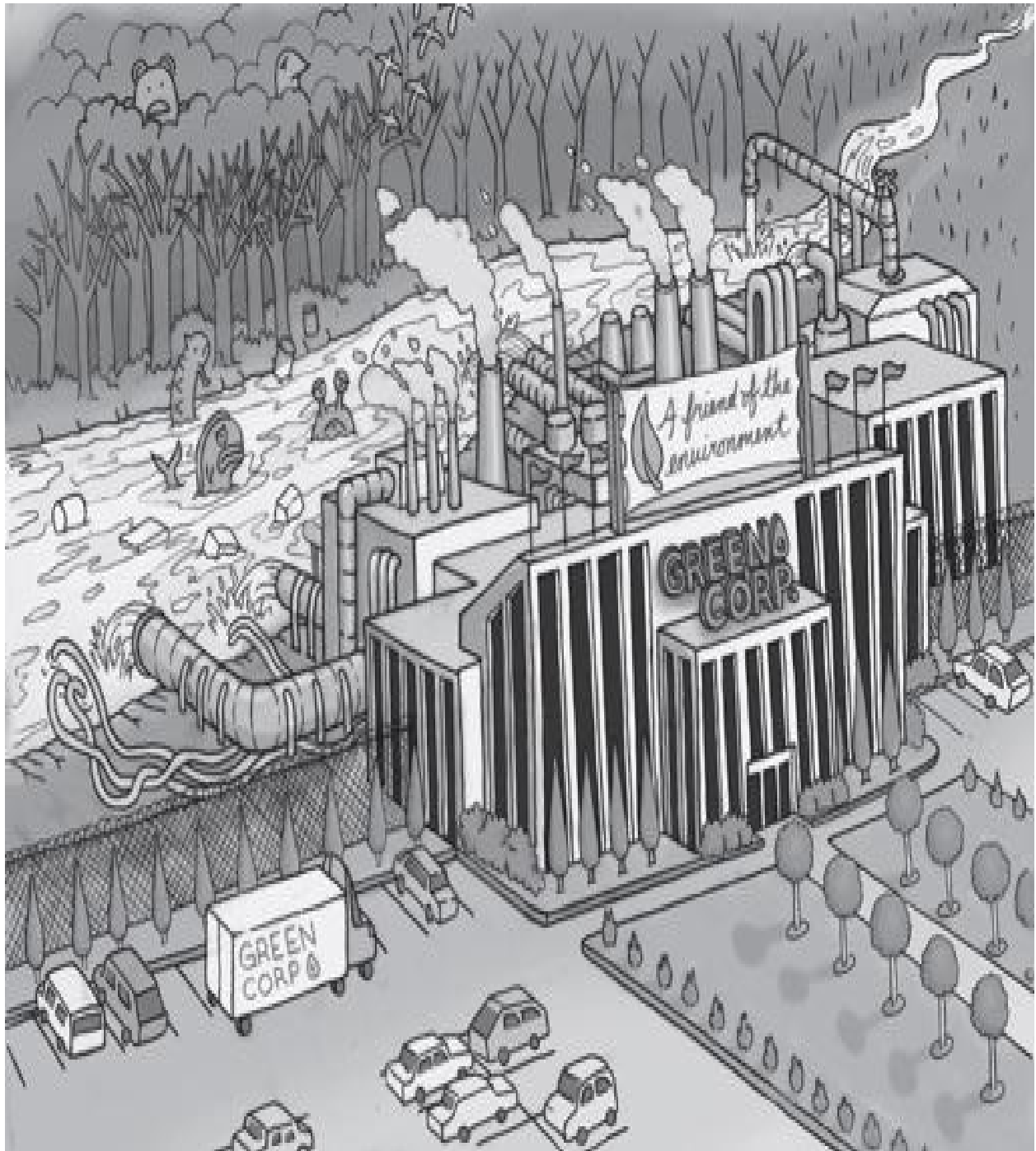
At the beginning of Paul O'Neill's tenure as CEO of Alcoa, he discovered a culture that rewarded executives with lavish perks, including private dining facilities, the best parking spots, access to a hunting lodge in Tennessee, and luxury apartments in New York City. These perks—like the free coffee and Danishes in the executive lounge—were off limits to regular workers. And that contradicted O'Neill's message to Alcoa staff that “everyone here is important.”

Saying one thing while doing another is a mixed message. Humans tend to believe that if someone's actions don't match their words, that behavior is bound to happen again. Mixed messages multiply mistrust.

So O'Neill removed Alcoa's mixed message by systematically eliminating executive perks. He bulldozed the hunting lodge and gave the property to The Nature Conservancy. He removed the special parking spots and closed the private dining facilities for executives. The luxury New York City apartments, which executives had used to take their families to Broadway shows, were sold. O'Neill eliminated all symbols and signs that said to people, “We're not all the same. We're not on an equal footing.”

Mixed messages exist in most organizations. Ask anyone if they've encountered one and it won't take them long to give examples.

What They Say	What They Do
“Quality is a core part of our mission.”	Ignore quality when it requires time and money.
“We value inclusivity, diversity, and equity.”	Fail to confront sexist and racist practices and policies.
“People are our most important asset.”	Treat staff poorly and neglect to invest in them.
“We care about the environment.”	Pollute regularly or deprioritize proactive steps to environmental stewardship.



Anyone who is in a position to lead or communicate values, or who aims to get better at something, is at risk of either giving or appearing to give a mixed message. I've promoted psychological safety¹ within my own company, and on a few occasions—particularly after some parenting challenges—I have not been a poster child for the practice. And just like that, a mixed message is born. I've worked hard to explain to my colleagues that while I'm 100 percent behind psychological safety, I'm also human and have room for improvement.

Sometimes, when you think your words match your actions, someone else sees it differently. When my company began following the guiding principle “Make Safety a Prerequisite,” everyone was asked to identify anything they saw that was unsafe. We ended up with a long list, which was a bit overwhelming. So we prioritized and began working on the highest-priority items first.

A little while later, an employee said that we weren’t serious about safety because we hadn’t gotten to one of his items yet. In his mind, we were sending a mixed message. In hindsight, instead of compiling a list of items and prioritizing them, it would have been better to leave it up to staff members to make their own decisions about what to improve.

Feelings of inequality and even fear may result from mixed messages. Kathleen Ryan and Daniel Oestreich, the authors of *Driving Fear Out of the Workplace*, describe an example based on their many interviews with workers:

A number of those interviewed commented on the fact that their managers rarely acknowledge them when they passed in the hall. This lack of common courtesy was taken as the sign of a caste system. When a manager behaves in a way that can be interpreted as impolite or deliberately distant, the message sets off a line of thinking that goes something like: “This person does not care about people. Which means that this person probably does not care about me. Which means that I may be in trouble.”²

It’s fair to assume that mixed messages exist in your world. In their book *Social Engineering*, Christopher Hadnagy and Paul Wilson observed the following about communication:

Never take for granted that the receiver has the same reality as you. Never take for granted that the receiver will interpret the message the way it was intended. Communication is not an absolute, finite thing. Always assume as many different realities exist as there are different people involved in the communication.

If you aim to create high-performance teams, working within an environment in which people operate at their full potential and trust one another, identify and eliminate mixed messages. As Paul O’Neill once said, “You have to constantly look for things that counter your message.”

Systematically identify and remove mixed messages.

¹. Amy Edmondson, “Psychological Safety and Learning Behavior in Work Teams,” *Administrative Science Quarterly* 44(2) (1999): 350–83. doi: 10.2307/2666999.

². Kathleen Ryan and Daniel Oestreich, *Driving Fear Out of the Workplace: Creating the High-Trust, High-Performance Organization* (San Francisco: Jossey-Bass, 2nd edition, 2008).



Gary V's ROI of Caring

On December 23, 2001, Gary Vaynerchuk's dad was pissed! It was two days before Christmas, the busiest day of the year in their liquor store, and a day on which they normally sell \$40,000 worth of liquor per hour. But Gary, who was loved and respected by customers and a critical part of the sales team, wasn't in the store!

Earlier that day, a customer had called to complain that the case of Beringer White Zinfandel she ordered hadn't been delivered and she desperately needed it for her upcoming Christmas Eve dinner. The entire fifteen-bottle case (not the normal twelve) only cost \$45!

After some debate about what to do, Gary grabbed the three-dollar-a-bottle case of wine and drove through the snow to the woman's house in Bergen County, New Jersey. When he arrived, the woman opened the door and said "Great!" She grabbed the case and shut her door.

By the time Gary returned to the liquor store, two-and-a-half hours later, his colleagues and father, who'd been selling liquor as fast as possible, were not happy that he had been gone for so long.

But Gary was pumped. He knew he'd made the right decision, and every salesperson in their liquor store would come to learn from his example. Gary said, "Over the next two to three years, that story became the foundation of how we treated every single customer—it became our competitive edge."

On that busiest day of the year for selling liquor, Gary says he used his heart, not his brain, to decide what to do. Truly caring about your customer, even your lowest-end customer, has a huge return on investment (ROI). Decisions like the one Gary made that day helped him and his dad grow their business from \$3 million in sales per year to more than \$60 million.

Care for people. Use your heart, not your brain.



Balanced Teams Perform Better

Two mathematicians from Queen Mary University of London already knew which team would win the 2010 FIFA World Cup before the game was even played. They'd made their prediction well known in advance, and it turned out they were absolutely correct. Was it dumb luck or were they really onto something?

To make their prediction, Dr. Javier López Peña and Dr. Hugo Touchette analyzed the frequency of passes between soccer players on all the FIFA teams and then mapped them.¹ These maps highlighted when one or more players on a team rarely handled the ball or rarely passed the ball back and forth with other players.

By analyzing the data, it was clear: Spain would win because their team was more balanced than their Dutch opponents. The mathematicians' passing map for the Dutch team revealed players who rarely handled the ball—an unbalanced team. By contrast, the passing map for the Spanish players indicated that there were no weak members on the team. Everyone trusted everyone else with the ball, which, it was predicted, would make a big difference in the final score. And it did.

To develop championship basketball teams at UCLA, coach John Wooden taught his players physical, mental, and emotional balance, as well as the balance of “losing themselves in the group for the greater good of the team.” And even that wasn't enough balance for the legendary coach, who also developed “squad balance, rebounding balance, offensive balance, defensive balance, and size balance.”²

So, what does this mean for you and me? It means that if we aren't “passing the ball” to certain players on our team, then our team is unbalanced and needs work. The right balance of people, participation, diversity of opinion, and creative tension helps a team achieve great things.

If you want better performance, strive for better balance in your teams.

¹. Javier López Peña and Hugo Touchette, “A Network Theory Analysis of Football Strategies.” In C. Clanet (ed.), *Sports Physics: Proc. 2012 Euromech Physics of Sports Conference*, Éditions de l'École Polytechnique, Palaiseau, 2013, 517–28.

². Wooden, *Wooden*, 1997.



Awful to Awesome at Yahoo!

In 2012, when Stas Zvinyatskovsky, Ed Kraay, and their team began to pull Yahoo!’s software development out of what they called a “tar pit,” teams averaged twenty weeks to build, test, stabilize, and deploy large software releases. Every release to Yahoo! customers was accompanied by painful and stressful production incidents, and programmers worked overtime—fixing defects, handling emergencies, and just trying to survive. As one developer explained, “We work really hard, but we don’t know if we’re making a difference.”

Long story short, Yahoo!’s competitors were running circles around them and something had to be done.

Two years later, it was normal to have daily releases and close to zero production incidents. Managers and employees across the entire one-thousand-person, geographically split organization, along with Yahoo! customers, were delighted by the improvements. It was a joyful ending to a painful journey. Stas and Ed described it as pulling an elephant out of a tar pit.

The tar pit of software development at Yahoo! was a software development process that was the exact opposite of agile. Defects leaked into production regularly, code changes led to even more defects, and no safety net of automated tests existed. Programmers worked solo for long periods of time, isolated from others’ code changes, which made integrating code slow and difficult. No automated builds existed, so manual builds had to be scheduled by a manager and would often break when code wouldn’t compile. After a successful build was finally produced, quality-assurance people would find serious defects in it. Programmers then fixed the defects, new builds were attempted, and the cycle repeated. The work was so slow and painful that most teams barely managed three releases per year.

Stas and Ed knew that serious changes were needed, yet it wasn’t obvious what to do. For more than a year, they tried many experiments, including creating a better testing environment, shifting from working on one big release to working separately on features, getting serious about a practice called *continuous integration*, and more. But the quality problems persisted, and teams *still* failed to get out more than three releases per year.

This is when Stas and Ed decided that they needed to fundamentally change the way they worked at Yahoo!. Instead of having a goal to be agile or to implement continuous integration, they realized their real goal was to “move fast to delight customers.”

To do that, they reasoned that they could no longer work in silos—one for product management, one for engineers, one for quality assurance, one for release engineering—where each silo threw work over the wall to the next silo. Instead, they would need to form balanced communities, the kind that could move from product idea to production without ever slowing down to ask any other team or department for help.

And that's exactly what they did.

The new balanced communities—forty to fifty people arranged in several smaller groups—focused on quickly and easily delivering delight to customers. In fact, my company, Industrial Logic, trained Yahoo!'s programmers during this period in a software design practice known as test-driven development, and we trained quality engineers in automated testing techniques. Every community now committed to automated builds, continuous integration, and continuous deployment: any new code changes that passed the automated build went automatically to production without any human intervention.

This was the breakthrough that Yahoo! needed to “move fast to delight customers.” The number of releases to production trended up sharply, while the number of production incidents trended steeply down. Working at Yahoo! became fun again. The elephant that emerged from the tar pit now moved with balance and grace.

The right people with the right skills committed to the right outcome could be your ticket out of a tar pit.



Balanced Teams Are CLEAR

You can follow the conventional wisdom about running successful meetings by keeping them short, with a clear agenda and minimal distractions. But if people don't feel safe to contribute their ideas, you will not benefit from their intelligence and creativity.

Patrick Lencioni, author of many great books, including *The Five Dysfunctions of a Team*, once tweeted this: "If someone offered me a single piece of evidence to assess the health of an organization, I would want to observe the executive team during a meeting."

In *Smarter Faster Better*, Charles Duhigg described Project Aristotle, Google's two-year study to identify the common traits of their most productive teams. Did the best teams have the right mix of people? Did they play together after work? Did they have the same hobbies or similar educational backgrounds? None of their observations matched a common denominator.

Then they came upon the work of Harvard's Amy Edmondson, whose 1999 study proposed that psychological safety was essential to high-performance teams. When Google researchers started looking for psychological safety on their most effective teams, they finally found an invaluable common denominator.

Have you or a colleague remained silent during a meeting with others? People who don't talk during meetings may feel like they aren't being heard or can't get a word in because others are dominating the discussion. Or they may fear saying something that would make them look ignorant.

After reading about Google's discovery, meetings at my company changed. We became far more aware of who wasn't talking and made space for them to speak. In addition, we made a point of becoming active listeners—regularly repeating or reviewing what people said so they felt heard.

During a virtual meeting, an individual in one of our remote offices hadn't said a word. Meanwhile, someone else was speaking a lot! During a brief pause in the discussion, I asked if our colleague in the remote office had anything to say. "Thank you, Josh!" he immediately replied. "I've been trying to jump into this discussion but was having a hard time doing so." He then went on to contribute some valuable ideas.

Experiences like this helped us validate Google's findings. Our meetings quickly became more *balanced* as we all endeavored to make sure that people had roughly equal time to speak and that they were encouraged to do so.

We also worked together to define five important tips for psychologically

safe meetings, derived from Google's findings and Charles Duhigg's writings. We organized these tips into an acronym and asked each other, "Can we agree to be CLEAR?"

Curious, caring, and open-minded

Listen to one another

Encourage everyone to contribute

Avoid dominating or interrupting

Repeat and review people's points

The above tips work wonders for leveraging human creativity and intelligence. They also help us achieve greater balance and grace.

Establish better balance and grace by leveraging everyone's intelligence and creativity. Be CLEAR!



Stop Interrupting People

While I was sitting in a meeting in the boardroom of a fast-growing company, I was pained to hear someone constantly interrupt somebody else. The person doing the interrupting was a male executive and the person being interrupted was a female director. After apparently tiring of being interrupted by her colleague, the woman remained quiet for the remainder of the meeting.

After the meeting, I asked her privately about being interrupted by the executive. She smiled and said, “Oh, yes, that happens all the time around here.”

Recently, a female colleague of mine interviewed a male candidate. He interrupted her so frequently during the interview that she decided to tally the number of interruptions. Over the course of thirty minutes, he interrupted her ten times! Now, we know that lags in internet connections can sometimes account for interruptions, but ten interruptions during one thirty-minute Zoom call is hard to reconcile. Needless to say, we didn’t hire this individual.

Interrupting women is the norm in the business world, and it’s a disgrace. Women who are routinely interrupted eventually remain silent, and silence is deadly when it comes to fixing problems, adding value, and innovating. A tremendous amount of value is lost when people are constantly interrupted. And it leads to churn, as those people eventually seek new employment.

The right balance of voices and perspectives is essential for organizational health. Stop interrupting people.



Etsy's Blameless Culture

What would you do if a new employee crashed your website and took it offline just days after joining your company? Would you fire them, penalize them, warn them, mandate training for them, or remove them from any work that could lead to such a disaster again?

Etsy, the online marketplace for arts and crafts, did none of those things. They operate under what they call a *blameless culture*. When failures happen, people don't blame other people, but instead use the situation as an opportunity to learn. The company regularly conducts *blameless postmortems*, a chance to safely reflect as a team on successes and how to learn from any failures.

Seven days into his new job at Etsy, an engineer deleted a file that he believed wasn't being used and pushed the change to production. Within minutes, [Etsy.com](https://www.etsy.com) went down. It turned out that the file was being used, but only for one specific web browser. When the production site tried to load the page, it needed to load contents from the deleted file. When that content could not be loaded, the production site tried to show an error page. Then it turned out that the error page itself could not be loaded because it, too, relied on the deleted file. This led to an infinite loop, which ate up all the memory on the servers. As a result, [Etsy.com](https://www.etsy.com) crashed.

So, what happened to this engineer?

He won Etsy's Three-Armed Sweater award, given out for the most surprising or spectacular accidental failure by an engineer. More important, he was thanked by engineers for revealing a really bad design. If an engineer deleting one file can bring down the company's entire website, the design needs work. The team conducted a blameless retrospective, came up with a more resilient design, and turned the failure into an opportunity to improve.



This engineer isn't afraid to make mistakes at Etsy because he knows he works in a genuinely blameless culture. As a result, he won't hold back from trying things—and maybe even occasionally breaking things.

Foster a blameless culture that treats people with dignity and respect. Use failures as opportunities to learn and improve.



The Blame Game

Myles, a twenty-two-year-old philosophy major who'd taught himself to code, had just made a really big mistake. He was working on an important new feature for a social media game played by tens of thousands of users. As a junior developer, he wasn't aware that he was working in an incredibly unsafe way.

While developing the new feature, Myles would delete data from a database table in production (i.e., on the computers used by paying customers) and have his new software re-create the table contents. In programming, it is extremely dangerous to work on production computers. Most developers use a database on a development machine so that if they make a mistake, it's not a problem to just re-create the database. This makes it safe to fail. But Myles didn't realize the danger of working on data in production. And since nothing bad had happened so far, Myles continued to work in this way.

Then disaster struck. One afternoon, while feeling drowsy, Myles brought up his tool for deleting the contents of a database table and accidentally deleted all the contents from the wrong table! As a result, everything in the user table, which stored all user identity and purchase information, instantly disappeared!

All hell broke loose. Customers couldn't log in, programmers scrambled to fix the problem (quickly learning that database backups hadn't been running for two months!), and executives called for an emergency meeting in the conference room.

"This is a monumental fuckup. You're gonna cost us millions in revenue," the CEO of the company raged at young Myles.

Myles worked with the rest of the staff around the clock for three straight days to fix the problem. All other work came to a halt. Since they had no full backup of the most recent user table, Myles and his colleagues had to call paying users to ask them what they had bought and manually enter the data back into the production database.

After the problem was fixed, Myles felt that coworkers treated him differently. He said, "I found myself weighed down with feelings of guilt and inadequacy." At a company all-hands meeting, he apologized to everyone. A few weeks later, he resigned from the job. After that, he gave up programming altogether.

Myles's company allowed people to work in highly unsafe ways. Their production database wasn't being backed up; their engineers had no safe, standard development process that would protect them from making huge, costly mistakes; people worked solo rather than in pairs or groups that could increase

safety; and a development sandbox didn't exist. When the CEO blamed Myles, he failed to recognize how unsafe his company was and how that contributed to a culture of blame.

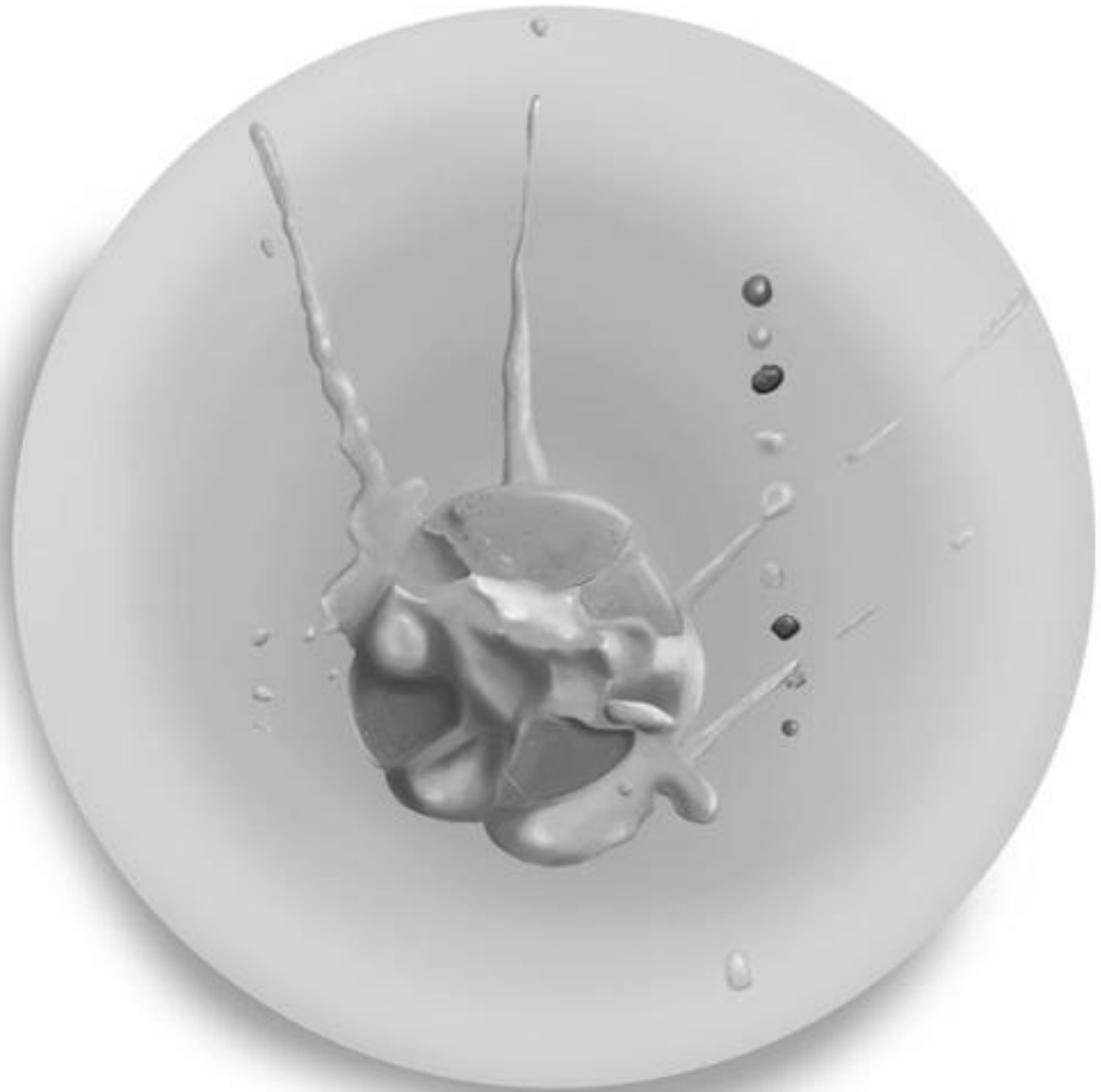
If you want high performance and professional results, treat your people with dignity and respect. Start by making it safe to fail.



Discover Art in Mistakes

Massimo Bottura owns and runs Osteria Francescana, a three-star Michelin restaurant in Modena, Italy. One day, his sous-chef was preparing two lemon tarts when, suddenly, one of the delicate tarts fell and broke apart. The sous-chef, an extremely experienced and highly skilled professional, felt awful. How could he be so clumsy? What would they do now? The diners were waiting for their dessert!

Massimo's response to his sous-chef's mistake echoed his philosophy that "mistakes are human, and in some way, they are beautiful." Instead of trying to quickly make a new lemon tart to replace the broken one, Massimo broke the remaining lemon tart, assembled the broken pieces on top of gelato, and splattered each plate with sauce. It looked as if the plates had fallen. He then named the new creation, "Oops! I dropped the lemon tart!" It was an immediate hit with the diners and went on to become an iconic dish at Massimo's restaurant.



In an interview years later, Massimo explained, “One of the most important things in life is to leave a free space for poetry.” Given the obligation to deliver dessert to those diners, Massimo gave himself the space to turn a mistake into poetry. He went far beyond “learning from failure.” He allowed himself to discover art in what others would only see as a failure. Imagine how wonderful his sous-chef felt when his boss helped transform a mistake into a win, and when that win went on to become the restaurant’s most famous dessert.

It is said that culture eats strategy for breakfast. The culture in Massimo’s three-star Michelin restaurant is joyous and artistic. It is safe to fail there because maybe your failure will become the next iconic dish. As James Joyce once said, “Mistakes are the portals of discovery.”

Discover art in mistakes.



Give Feedback Gracefully

Felix is picking his nose and it's totally disgusting," several freaked-out team members told me. Felix had only recently joined the team, and I was one of several coaches helping them to adopt agile development.

The team's space was divided into commons and caves. In the commons area, team members collaborated on shared workstations. In the caves area, everyone had their own small, individual place where they could store belongings and do some solo work. Since the team spent most of its time in the commons area, and Felix was using shared keyboards and mice, something had to be done!

I thought about the problem and decided that Diana Larsen would be best suited to handle it. Diana is a deeply experienced agile coach, organizational designer, author, and gifted people person. She decided not to approach Felix directly because it was really Felix's manager's job to give his employees feedback. Diana met with the manager, explained the situation, and then offered to run a team-improvement session during which people, in pairs, could give each other helpful and respectful feedback. She would arrange for Felix to get paired with his manager so the manager could give Felix the feedback.

It worked perfectly. The manager gave Felix the feedback about his nose picking. Felix was surprised and embarrassed. He went for a walk to get some air. When he returned, Felix told his manager that he had absolutely no idea that he was picking his nose, that it was an unconscious thing and that he would work extremely hard not to do it any longer. The manager was happy, and as the days passed, the team was deeply relieved that Felix had stopped picking his nose.

I was also relieved that I didn't have to personally handle this delicate and awkward situation. But a few years later, I was confronted with this exact same issue, in my own office! Sure enough, someone who worked for me was picking his nose! Now it was *my* turn to have the difficult conversation.

I asked the individual to go for a walk, and I gave him the feedback once we got out of the office. He also said he wasn't aware that he was picking his nose and said he'd stop doing it. And he did, mostly. Occasionally he'd still pick his nose, but it was infrequently enough that I didn't need to say anything more about it.

Don't avoid dealing with behaviors that disrupt a team's ability to collaborate. Always give feedback gracefully.



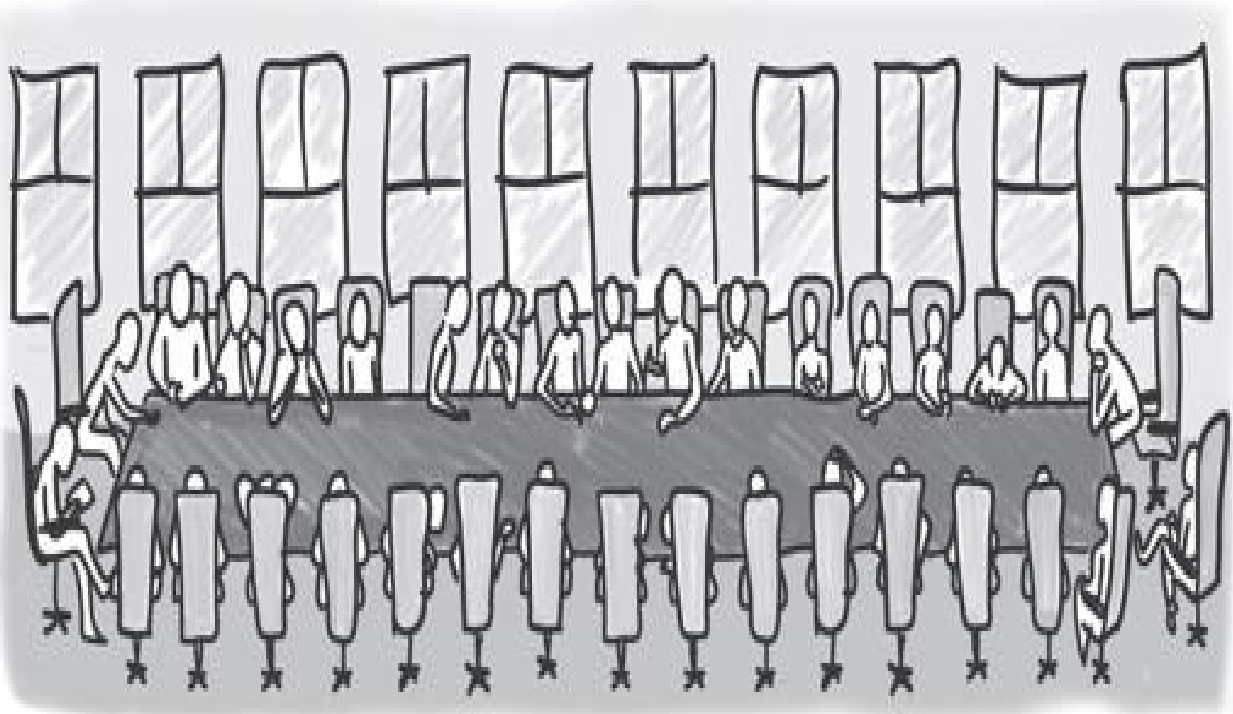
The Emperor's New Table

The largest conference table I've ever seen was in a cavernous boardroom at a pharmaceutical company in Basel, Switzerland. It was a giant rectangular table, surrounded by luxurious leather chairs. Employees told me that this room made them feel quite uncomfortable, however, since it was used to approve or deny drug development plans—a process that they said felt a lot like an interrogation. These drug plans were refined by many people over many months and subject to strict design guidelines. If you failed to follow the guidelines, your development plan was immediately rejected.

The room had no warmth, and because no other training rooms were available, I had to teach a workshop there. As you can imagine, it was a less-than-ideal setting for a workshop that involved lots of collaboration between small groups of people.

Tables can either foster or inhibit collaboration. Ed Catmull, cofounder of Pixar and former president of Walt Disney Animation Studios, explained in his excellent book *Creativity, Inc.* how Pixar was unaware of a collaboration problem with their own conference table for thirteen years! The elegant table was wooden, rectangular, and quite long. It had been chosen by a designer whom Steve Jobs admired, which led people to ignore its shortcomings and accept the table—warts and all. Explained Ed:

We'd hold regular meetings about our movies around that table—thirty of us facing off in two long lines, often with more people seated along the walls—and everyone was so spread out that it was difficult to communicate. For those unlucky enough to be seated at the far ends, ideas didn't flow because it was nearly impossible to make eye contact without craning your neck.¹



Ed notes in his book that Pixar is a place where they respect people and expect everyone to speak up, regardless of title or role. One day, in a smaller room, he noticed that collaboration was far better around a square table:

Sitting around that table, the interplay was better, the exchange of ideas more free-flowing, the eye contact automatic. Every person there, no matter their job title, felt free to speak up. This was not only what we wanted, it was a fundamental Pixar belief: Unhindered communication was key, no matter what your position. At our long, skinny table, comfortable in our middle seats, we had utterly failed to recognize that we were behaving contrary to that basic tenet.²

Soon after that meeting, Ed immediately asked facilities to get rid of the long, elegant rectangular table. Like any organization, Pixar makes mistakes. Yet, if you read Ed's book, you'll discover that they learn from their mistakes and constantly strive to walk the talk of their values.

Similarly, the pharmaceutical company no longer holds drug development planning meetings in that cavernous room with the enormous table. They now take a more iterative, agile approach to drug planning, with less focus on document formatting rules, a greater appreciation for evolving a plan, and more frequent meetings in smaller, more collaborative spaces.

Never let a poorly designed space hinder collaboration or work against your values.

¹. Ed Catmull and Amy Wallace, *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the*

Way of True Inspiration (New York: Random House, 2014).

[2](#). Ibid.



When the Plans Came Tumbling Down

The staff of a Nielsen division in Florida got to work early one morning and were stunned to find that *all* of their agile planning posters were gone. The posters, many of which contained handwritten index cards and estimates on Post-it Notes—reflecting countless hours of teamwork—were nowhere to be found. As more people arrived at work that day, the outrage was palpable.

Who could have done this?

Some wondered whether the facilities department removed the posters, but people in facilities had become used to the posters and knew not to mess with them.

Others suspected a few individuals on teams that hadn't yet switched to agile and who were hostile to the large-scale agile adoption initiative.

After the division had completed a few successful agile pilot projects, its leadership team decided to go all-in on agility. Industrial Logic had helped from the beginning and continued to do so for this large-scale adoption. But when leaders announced plans to spread agility across the entire division, some were not happy. Did those unhappy people tear down the planning posters?

Katherine was the division leader. She was a no-nonsense VP who knew that her group needed to improve. The first agile pilot projects—which helped her staff learn and practice technical and managerial agility—proved that she was right. Most everyone was happier and far more efficient and effective with agile. More than a year and a half into their agile adoption, Katherine knew it was time to go division-wide.

But then the planning posters came tumbling down.

Katherine called an all-hands meeting. She explained that they had investigated the incident and found nothing. Then she said something that I will never forget: “We know that none of you in this division would have ever removed the planning posters, so we are simply going to move forward and continue with our agile adoption.”

That was it. No one was accused of foul play. No one was disciplined. No one was fired. Teams simply picked up where they had left off. They made new planning posters and occasionally took photos of them, just to be safe. The agile adoption continued across the entire division and results were so good that another division began its own agile adoption.

When faced with an unexpected act of sabotage, keep calm,

assume innocence, avoid blame, and continue moving forward gracefully.



From Sith Lord to Superboss

Tim Ottinger, a lean/agile guru, was once asked to coach a manager with a horrible reputation. He learned, via people who reported to this manager, that he was overbearing, micromanaging, angry, semi-distracted, impatient—all the things you don’t want in a boss. Tim noticed that the manager seemed to have a dark cloud constantly hovering over him. Some people decided it was a cloud of evil, like a Sith Lord in *Star Wars* or something. They avoided talking to him and whispered behind his back when they talked about him.

This wasn’t a general labor versus management issue. Most of the people at this company loved their managers, and *love* wasn’t too strong a word. There was respect, admiration, comradery. That made the experience doubly frustrating for the people who had previously reported to one of the company’s beloved managers and now worked for this awful manager.

Tim was asked to coach middle management on team leadership. He was given a set of one-on-one meetings with several managers, including the Sith Lord. Based on how troubled the manager’s direct reports were, Tim decided to exercise “curiosity over judgment” by first learning more about the manager via *his* boss.

He learned that the troubled manager was from Europe and was still learning American culture and behaviors. This was his first managerial position, the result of a promotion. Finally, Tim learned that while he was an inexperienced manager, his boss and others in the organization believed he was of good character and intellect, and they had a lot of faith in him.

Tim analyzed the manager’s work and realized that the problem he had was trying to manage and create one-on-one relationships with fifty to sixty individual developers. It was simply too much work. Tim suggested to the manager that rather than directly dealing with fifty-plus people, why not start by addressing teams of developers and then let the relationships form slowly over time? If he could treat each team as a “team-person,” then he could assign work and follow up with five teams instead of fifty people. He could follow up on the work rather than on the individual. Then his one-on-ones would not be focused on status and completion, but on a more human basis of intentions, goals, and professional development opportunities.

After this discussion with the manager, Tim left for a few weeks. When he returned, the team members cornered him immediately, asking, “What did you do to our boss?!? He’s acting totally differently and we don’t trust him.”

“What do you mean?”

“He’s been personable, nice, even human! He has completely stopped micromanaging us and gives us a lot of autonomy and latitude so we can finally get our work done! We don’t know if it will last, but it’s amazing!”

When Tim last inquired, this once-horrific micromanaging monster had become a trusted and respected boss and was moving nicely through the ranks on the basis of his teams’ productivity.

It is well known and well understood in agile software circles that “the team is the unit of accountability.” It’s not a unique concept or rare insight.

The move to managing teams instead of individual workloads transformed the manager’s practice in a positive way. He was able to unlock both his potential as a manager and his teams’ potential to grow and achieve.

There’s a superboss lurking inside of every Sith Lord. Exercise curiosity over judgment to help unearth a person’s true potential.



Change by Attraction

A software developer once learned that a Kanban board can be a useful tool for visualizing and managing the state of a team's work (e.g., what work was started, in-flight, or completed). He mentioned it to his teammates, but didn't get much interest. So he started tracking his own work with a Kanban board. Soon, someone noticed and asked some questions. The developer explained how it helped him, but didn't push it. The other person decided to try tracking his own work using a Kanban board. Soon there were three people using personal Kanban boards. When the developer brought it up again to the team, there was enough momentum that the rest of the team was willing to try it.

Esther Derby—management guru, lean/agile expert, and author of several excellent management books—told me the above story. It's a wonderful illustration of three agile mantras: Start Minimal and Evolve, Be Readily Resourceful, and Be Balanced and Graceful. Esther uses this story to describe what she calls *change by attraction*. In her book, *7 Rules for Positive, Productive Change: Micro Shifts, Macro Results*, Esther explains that change via “pushing, persuading, cajoling, and sanctioning” is often met with resistance, whereas change by attraction “has the opposite effect: resistance fades because there is nothing to push back against; there is only something to move toward, by choice.”

Moving toward a change by choice, rather than by decree, is often more effective and graceful. In my agile consulting work, I encounter managers who want their teams to improve. Rather than impose changes on these teams, it's usually better to start by learning what is or isn't working well. By understanding the challenges people face, it becomes easier to invite them to experiment with agile practices that may alleviate their stress and make work more enjoyable.

In my consulting work, after showing team members videos of agile practices and sharing agile stories, I ask each team member to take an anonymous poll. On a scale of 1 to 5, how excited are they to experiment with several new agile practices? If I don't get a sufficient number of 4s and 5s, I don't begin the work.

Change is more graceful when people are empowered to make decisions about their own experiments. Esther also makes a point of saying that change by attraction isn't always the right strategy, and she provides additional approaches for positive, productive change. I highly recommend studying her book.

It's usually ungraceful to impose change. Try change by attraction

instead.



Summary: Be Balanced and Graceful

Being balanced and graceful is a key to achieving what Paul O'Neill called "habitual excellence." To do that, you must care deeply for people. Begin by systematically identifying and removing mixed messages. Caring for people requires using your heart, not your brain.

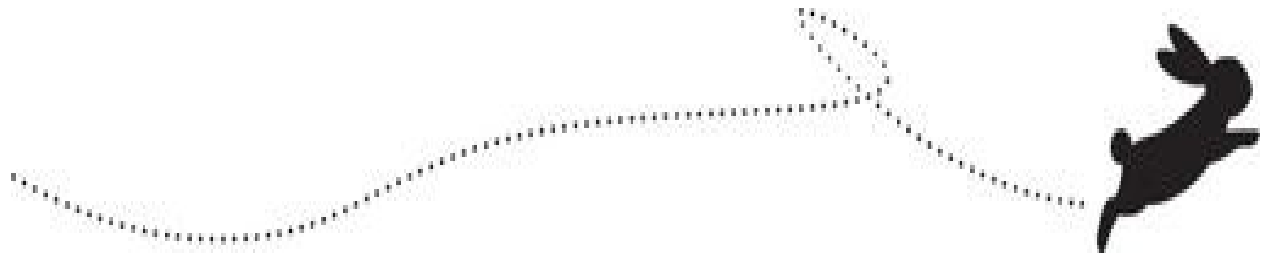
To achieve better performance, strive for better balance in your teams. A balanced team has the right people with the right skills committed to the right outcome. You will establish better balance and grace by leveraging everyone's intelligence and creativity.

Giving people a voice and not interrupting them is essential for organizational health. Foster a blameless culture that treats people with dignity and respect. Start by making it safe to fail. Use failures as opportunities to innovate and improve. Discover art in mistakes.

Don't avoid dealing with behaviors that disrupt a team's ability to collaborate. If a poorly designed space is hindering collaboration or working against your values, empower people to fix it.

Always give feedback gracefully. When faced with an unexpected act of sabotage, keep calm, assume innocence, avoid blame, and continue moving forward gracefully. Remember that there is a superboss lurking inside of every Sith Lord. Exercise curiosity over judgment to help unearth a person's true potential.

Change is more graceful when people are empowered to make decisions about their own experiments. It's usually ungraceful to impose change. Try change by attraction instead.



Be Poised to Adapt

On one hand, being poised means being in balance or at ease. On the other hand, it also means being ready to act or move. When we are poised to adapt, we are ready to respond quickly and easily to changes. Examples include rapidly altering your approach based on new information, quickly modifying something for a new purpose, or adjusting rapidly to a new reality.

David D. Woods, a pioneer of resilience engineering, has studied the relationship between human coordination with automated and intelligent systems for more than forty years.¹ He explains that being poised to adapt is a readiness to revise and readiness to respond, which must exist before specific challenges are faced. This requires what he calls *adaptive capacity*: a system's readiness or potential to change how a system currently works—its models, plans, processes, behaviors—to continue to fit changing situations, anomalies, and surprises.²

Learning is vital for developing adaptive capacity. It prepares us to face the unexpected and gives us the ability to adapt rapidly. And as we learn and adapt continuously, we eventually either hit our goal or pivot to a better one.

When you become poised to adapt, it helps to consider the mantra Be Quick—But Don't Hurry. Don't rush learning, and don't hurry to adapt. This will help you avoid serious errors.

Learning often involves failing for a time and learning from what went wrong. Safety helps make that possible. If it isn't safe to fail, it will be difficult to learn.³ The mantra Drive Out Fear applies here.

Anything that slows down learning or interferes with our ability to adapt is potentially something worth reducing or eliminating. If you are developing a new product or trying out a new approach, what are safe, effective ways to learn efficiently so you can adapt rapidly and improve? Agile people find ways to get feedback from many sources, rapidly and safely.

The mantra Start Minimal and Evolve enables us to be poised to adapt. Begin with a basic, primitive version of something, quickly learn from it, adapt it based on what you learn, and repeat until you've attained your desired outcome.

¹. For an excellent example of this work, see the Stella Report (2017), snafucatchers.github.io

2. David Woods, “Resilience Is a Verb.” In B.D. Trump, B.-V. Florin, and I. Linkov (eds.), *IRGC Resource Guide on Resilience (vol. 2): Domains of Resilience for Complex Interconnected Systems*. Lausanne, CH: EPFL International Risk Governance Center, 2018.
3. Edmondson, “Psychological Safety,” 1999.



Virginia's Impromptu Solo on the D String

My friend Wyatt Sutherland, a lean/agile guru, professional cellist, and teacher, once told me a story about a virtuoso named Virginia Schneider, who was the principal violist in the Louisville Orchestra:

I was performing with the Louisville Orchestra, when the solo violist's A string broke in concert and she was forced to play the last part of Strauss's Don Quixote on the D string—an amazing feat since the passage was already high on the A string and to play the same notes on a lower string was Herculean. No musician would ever expect this to happen, but if the soloist hadn't spent a lifetime practicing her etudes, scales, and exercises she would not have had the chops to lean on in that unexpected moment.



Virginia's flawless performance resulted from being poised to adapt in an unexpected situation. While she had never rehearsed her solo on the D string, she performed it gracefully, thanks to years of practice and mastery of her instrument. She possessed high adaptive capacity.

Virginia was an accomplished violinist in her teens, studied at Juilliard and other prestigious music schools, taught students in Ohio and Kentucky, learned the Suzuki method from Dr. Suzuki himself in Japan, and founded the University of Louisville Suzuki String Academy.

As a young cello student in Virginia's Academy, Wyatt remembered learning

from Virginia to “strive for excellence no matter how far the bar was beyond our current ability, and while perfection was an imaginary destination, what truly mattered most was a daily commitment in pursuing excellence.”

Attaining excellence enables professionals to be poised to adapt. It is a pathway to agility and not generally something that happens quickly or easily. That’s why practice and mastery of one’s craft is so vital.

Being poised to adapt takes practice. Agility is an outcome of mastery. Commit to a daily pursuit of excellence.



How NASA Prepared for the Unexpected

After an oxygen tank outside of Apollo 13 blew up, the spacecraft's mission suddenly pivoted from landing on the moon to saving the lives of the three astronauts onboard. Ken Mattingly was supposed to be one of those astronauts, but an illness before the launch had sidelined him. Now he would assist mission control in rescuing the Apollo 13 crew.

Popular Science interviewed Ken about the famous 1970 mission. Ken explained that after the oxygen tank blew up, the command module (CM) began losing oxygen, which meant that the CM's fuel cells, life support, and pressurization systems would eventually quit. To survive, the crew needed to move into the tiny lunar module (LM).

Once inside the LM, the astronauts began their journey home by first circling the moon. Just as that part of the journey was completed, the CO₂ levels in the LM became dangerously high. The crew fixed the problem by replacing an air filtration canister. But the LM had a limited supply of those canisters, and once they were used up, the crew could not survive. The story of how NASA solved the CO₂ problem illustrates another important lesson in agility.

The CM also had an air filtration system, but it used square canisters. The LM system used circular canisters. The astronauts needed to make the CM's square canisters work with the LM's circular design. In the Hollywood movie about Apollo 13, a NASA engineer tells the flight director, "This just isn't a contingency we've remotely considered." What follows is a dramatic scene in which a bunch of parts (all of which are also onboard Apollo 13) are thrown onto a table for several NASA engineers to try to figure out how to make the square canisters work with a system designed for round canisters. The engineers do eventually figure out a solution that saves the lives of the Apollo 13 astronauts.

Yet, the real story of Apollo 13 didn't involve the spontaneous act of improvisation depicted in the film. Survival in the LM had already been simulated and even nicknamed the "LM lifeboat" option. Ken remembered it:

As part of our preparation for these missions, the simulation supervisors injected various anomalies and situations we weren't planning for [into the flight plan exercises] to test the contingency and emergency procedures we had developed.

On an early mission—it could have been Apollo 10 or Apollo 9—they

created a situation where they had contaminated the atmosphere in the CM, and they had to figure out how to get it out of there. They said, “Well, why don’t we move into the LM? And we’ll close up and depressurize the CM, get rid of all the cabin atmosphere, then repressurize it, open up the LM, and bring the crew back?”¹

To make the LM lifeboat option work, the NASA team needed to handle the exact situation of adapting square canisters to work with a round filtration system. When the Apollo 13 emergency occurred, the team recalled what they had done during the LM lifeboat simulation. Said Ken,

We cut open some plastic bags that we packaged stuff in, and using just plain old gray duct tape, taped a bag around the canister. Then we inserted it in the suit hose and taped it to one of the nozzles on each end, so you could blow the air through it. It would take some time to run it, but it would run just like if you put it in the manifold itself. We did the same thing [for Apollo 13].²

In other words, NASA had not only considered this contingency but had rehearsed what to do if it became necessary. Ken said,

The beauty in this whole thing was, these guys were so prepared for even the most implausible things. They knew no one had ever simulated exactly what happened, but they had simulated the kind of stress that could be applied to the system and the people in it. They knew what their options were, and had some ideas already in place about where to go.³

While the curveballs or anomalies you face will likely differ from something as dramatic as a NASA space mission, your ability to handle unexpected events with quick, easy grace, and to be resourceful and adaptable in finding solutions, will require practice and preparation. Benjamin Franklin once said, “By failing to prepare, you are preparing to fail.”

What potential problems might you encounter? What could you simulate in order to be better prepared for unexpected events? What safety measures could you put in place? What could make things more fault-tolerant or resilient?

Practice how you respond to problems. Prepare for the unexpected.

¹. Neel V. Patel, “The Greatest Space Hack Ever,” *Popular Science*, October 8, 2014, www.popsci.com/article/technology/greatest-space-hack-ever/.

². Ibid.

³. Ibid.



A Personalized Purchase

I needed new running shoes. I'd seen a store devoted to runners and thought I'd pop in to buy a new pair. I figured I'd be done in twenty minutes or less, but I had no idea what was about to happen.

This store didn't just try to sell me a pair of shoes as quickly as it could and send me on my way. They took the time required to customize the fit of the shoes to my feet.

The process started with a 3D scan of my feet, which was used to create right then and there—in less than five minutes—a set of custom insoles molded to my feet. Next, the salesperson had me run on a treadmill to see exactly how my feet landed and to look for misalignments in my ankles and knees. The salesperson spoke with me at length about the kind of running I'd be doing and used all of the data collected to recommend my ideal pair of shoes.

An hour after entering that running shoe store, I emerged with the best-fitting running shoes I'd ever worn. In addition, I was delighted that a store would go to such lengths to help produce an awesome outcome for my running. I love those running shoes, and I am now a devoted customer.

You will have happier customers if you can rapidly adapt your products/services to their exact needs.



Multiplicity and Selection

Imagine a *National Geographic* magazine with seventeen gorgeous photos of African elephants. How many photos do you think were originally taken to get to that final selection of seventeen? One hundred? A thousand? Guess again—the correct number is closer to thirty thousand!

Multiplicity and selection means finding ways to inexpensively produce many results from which you choose your best. In his book *Experimentation Matters*, Stefan H. Thomke described how BMW uses this approach to design lifesaving crumple zones in their vehicles. They perform millions of simulated crashes using a computer program to determine the ideal designs, metal densities, and compounds for doors, bumpers, and so on. This is light-years ahead of older approaches that required a great deal of guesswork and a paucity of data extracted from expensive, destructive tests with crash test dummies.

Management guru Tom Peters said in his book *The Excellence Dividend* that “whoever tries the most stuff wins.” Thomas Edison said, “The real measure of success is the number of experiments that can be crowded into twenty-four hours.” A founding philosophy of Richard Sheridan, cofounder and CEO of Menlo Innovations and author of *Joy, Inc.*, is to “make mistakes faster.” Ralph Waldo Emerson said, “All life is an experiment. The more experiments you make the better.” And Jeff Bezos said, “Our success at Amazon is a function of how many experiments we do per year, per month, per week, per day.” In 2016, *Fast Company* reported the following number of experiments conducted each year by innovative companies: Intuit (1,300), Procter & Gamble (7,000–10,000), Google (7,000), Amazon (1,976), and Netflix (1,000).¹

In his book *Originals*, Adam Grant wrote, “If you look at Beethoven, Bach, Mozart, it wasn’t that their average is so much better than their peers. It’s that they generated sometimes 600 or 1,000 different compositions. A few of those are considered true masterpieces.” A multiplicity of data, artifacts, or experiments can increase your chances of producing masterpieces. But how much is enough, since this work is not without cost?

Amazon faced this problem in its early days. Employees were running lots of experiments, many of which lacked a clear hypothesis and value proposition. Jeff Holden, who designed experimentation engines at Amazon, Groupon, and Uber, helped Amazon pivot to a model in which experiments were first reviewed by an experiments group. That group guided experimenters to check the validity and value of their experiments. This adjustment helped the company focus on valuable, instead of aimless, experimentation.

Most organizations struggle with the opposite problem: not experimenting enough. Consider Cheerios, the best-selling breakfast cereal by General Mills. Decades ago, an employee suggested adding a honey nut coating to Cheerios. The idea was dismissed. Five years later, the company tried the idea and quickly learned that test subjects loved it! Honey Nut Cheerios went on to become a massive best seller.

Focus on a clear value proposition and then use multiplicity and selection to learn and adapt rapidly.

¹. Ben Clarke, “Why These Tech Companies Keep Running Thousands of Failed Experiments,” *Fast Company*, September, 21, 2016, [fastcompany.com/3063846/why-these-tech-companies-keep-running-thousands-of-failed](https://www.fastcompany.com/3063846/why-these-tech-companies-keep-running-thousands-of-failed).

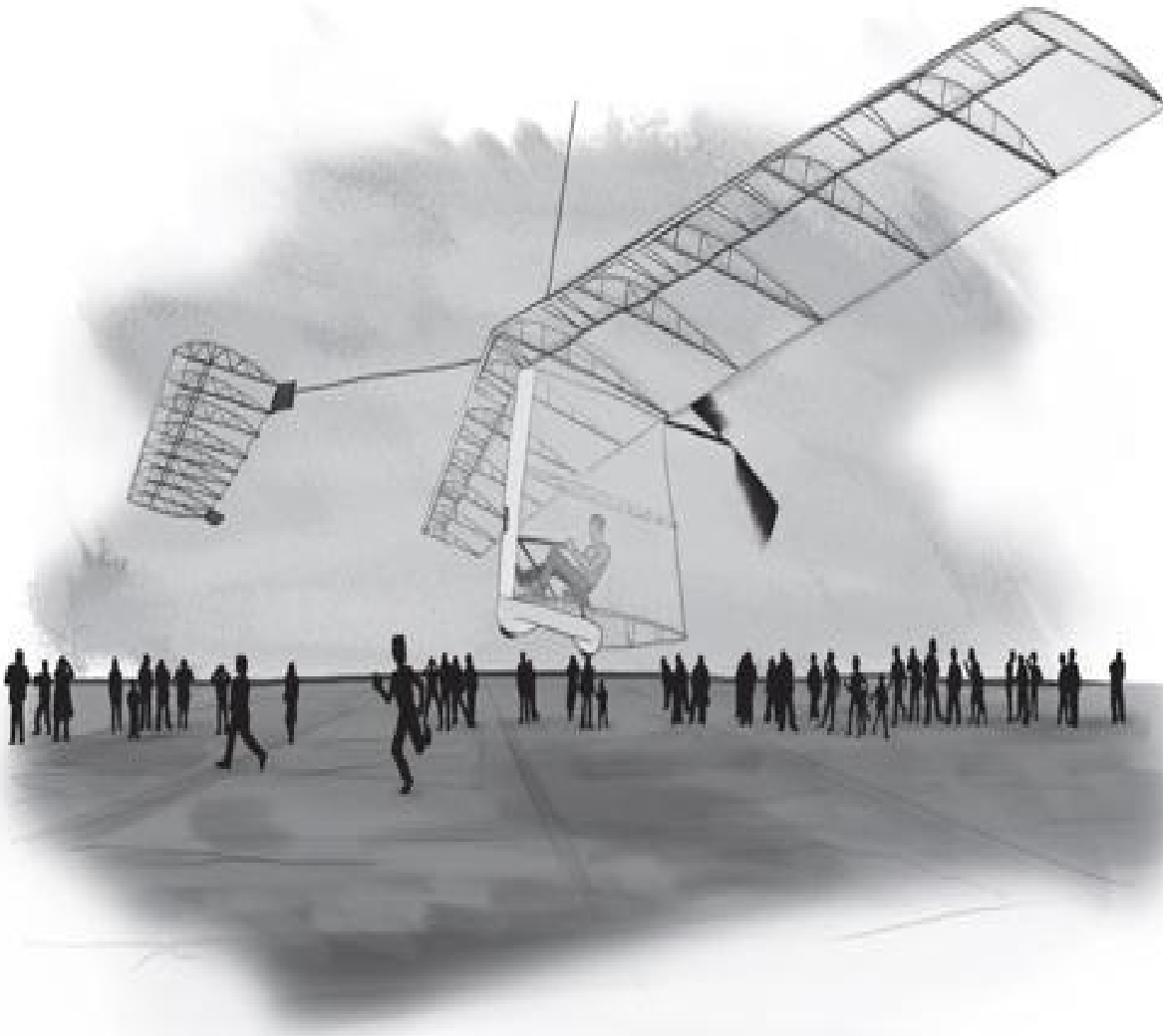


The Gossamer Condor

In 1959, Henry Kremer, a wealthy businessman, offered a large cash reward to anyone who could pilot a human-powered aircraft around a one-mile, figure-eight course. For eighteen years, no one succeeded. Then Paul MacCready entered the challenge. He considered what had been previously tried and declared, “The problem is we don’t know what the problem is.”

To get to an answer, MacCready engineered a process by which he could iterate safely and rapidly on the problem. He and his team used aluminum tubing, mylar, and wire to quickly produce experimental airplanes. The parts on these airplanes were ultra-lightweight and easily breakable—by design. In fact, if a part didn’t break, it was usually a sign that it was too robust and needed to be changed.

MacCready’s airplanes flew so slowly and so close to the ground that crashes were safe and fixing broken parts was usually quick and easy. Iterations of the airplane were in fact so safe that MacCready’s own son piloted many early versions. And whereas his competitors took weeks or months between test flights, MacCready and team attempted flights, failed, learned, adapted, and experimented again—all in a matter of hours or days.



MacCready and his team made 222 test flights over a yearlong period and attempted to win the Kremer prize nine times. On August 23, 1977—the team’s 223rd flight, and tenth attempt at the prize—the *Gossamer Condor* succeeded.

Paul MacCready is widely considered to be one of the greatest engineers of the twentieth century. Safety was integral to his success. But there is more to the story. Others in the competition for human-powered flight also focused on safety, particularly the safety of the pilot. The difference was that MacCready and team realized that safety was integral to adapting rapidly, and adapting rapidly was required to win the competition. Compared with his competitors, MacCready’s experimental airplanes *broke better*, requiring only quick fixes to get them back into the air, where learning resumed.

Being poised to adapt is key to being agile because it enables a safe, effective way to learn how to succeed sooner.

Fail better, faster, and cheaper to succeed sooner.



Iterative Experimentation

In 1986, Hirotaka Takeuchi and Ikujiro Nonaka authored “The New New Product Development Game,” a seminal article in the *Harvard Business Review* that associated the rugby term *scrum* with product development. They observed that slow, sequential approaches to building products would no longer suffice in a highly competitive world. Instead, businesses needed a new product development process focused on flexibility and speed.

These wise gentlemen from Japan described a less rigid way of working, in which a “multidisciplinary team” collaborated (instead of individuals working solo) to conduct “iterative experimentation” as they evolved a product together. It would not be unusual for such a team to “reconsider a decision as a result of later information.”

Takeuchi and Nonaka could have easily been describing the way MacCready and his team evolved the human-powered *Gossamer Condor* aircraft in 1977. The time between *Gossamer* experiments varied from hours to days, and sometimes even weeks. They were learning and adapting rapidly via iterative experimentation.

In 1995, two software consultants named Jeff Sutherland and Ken Schwaber formulated Scrum, a lightweight process—including roles, responsibilities, and rituals—based on Takeuchi and Nonaka’s ideas. These consultants introduced something the authors had never suggested: a short, fixed-length time box called a *sprint*. The idea came from MIT’s Media Lab, which had a policy that after every three weeks of work, teams had to demonstrate to their colleagues what they were working on.¹ This enabled lab directors to review progress and decide whether to kill a project. These “sprints” (a team working within short, fixed-length time boxes) in Scrum established a cadence for planning, working, and reviewing progress.

The idea was better than working for many months without reviewing any progress. And while the authors of Scrum never mentioned Takeuchi and Nonaka’s “iterative experimentation,” they described an “empirical process” that enabled learning and rapid adaptation by performing experiments inside of sprints.

Unfortunately, as Scrum gained in popularity, experimenting and learning rapidly didn’t remain the focus. My colleagues and I were often called in to help struggling Scrum teams. We would consistently find people on teams hurrying to complete the work they had estimated to fit into each sprint. Rather than finding an agile way to achieve an important outcome, they had become fixated on

predictability, performance to plan, and adherence to Scrum's roles, responsibilities, and rituals.

To make matters worse, many teams that adopted Scrum continued to work in their old, slow, sequential ways by simply dividing the work up into a series of sprints and giving work items to individuals rather than the whole team. Such misguided adoptions of Scrum, which are not uncommon, become “activity without achievement”—a different way of working that often fails to improve outcomes.

Takeuchi and Nonaka never suggested working in sprints or fixating on how much work a team could predictably complete from sprint to sprint. MacCready himself was terrible at estimation—he predicted he would be able to fly in two months and instead it took a whole year.

While Takeuchi and Nonaka never used the word *agile*, what they communicated in their brilliant article was the competitive necessity for product teams to be poised to adapt by working in ways that enabled speed and flexibility. Iterative experimentation conducted by a multidisciplinary team (called a “balanced team” in this book) enables such work and is a successful recipe for agility.

Being poised to adapt is easier with a balanced team engaged in
iterative experimentation.

¹. Jeff Sutherland and J. J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time* (New York: Penguin Random House, 2014).



Fail Smarter via Quick, Cheap Experiments

In 2010, Paul Howe theorized that people could make better product buying decisions by leveraging their social network. A Silicon Valley venture capital firm invested \$2 million to fund his startup. At the same time, two other entrepreneurs in the Valley also realized the potential of marrying social media with product reviews, and each got \$10 million in funding from the best VC firms.

All three of these startups would fail within a year. What's particularly valuable from their stories is *how* they failed.

The two startups that had attracted the \$10 million investments spent every dollar hiring senior engineers and furiously coding a fully featured service over nine months. When they launched their services, however, they quickly discovered that no one wanted to use them. Said one of the CEOs in TechCrunch, "We don't think people want to share their purchases. Period."

Paul's startup failed differently. It failed *smarter*. As a practitioner of the ideas in Eric Ries's book *The Lean Startup*, Paul understood the need to use a fast, capital-efficient way to validate or invalidate ideas. How could he quickly fake the social buying experience without spending a lot of money? He realized that a tool called Greasemonkey could help.

A Greasemonkey script can alter the contents of a web page running on a single computer without actually changing the real web page on the internet. Paul found a Greasemonkey programmer in the Philippines on RentACoder.com and paid him \$40 to write a script that could *simulate* a key part of his future service.

Paul wanted to learn whether people would like to see items in their Facebook newsfeed about what friends in their social network bought. Since he didn't want to alter anyone's actual Facebook newsfeed, the Greasemonkey script needed to automatically insert fake shopping items into the newsfeed that appeared only in the web browser running on the research machine. The script transmitted no data to the internet and didn't touch the user's actual social media account. Anyone who agreed to participate in this research would learn, immediately following the experience, that they'd seen some faked shopping items on the research computer.

The altered newsfeed on the research machine looked like this:



Test Subject

[View My Profile](#)

News Feed

Messages

Events

Photos

Friends

Applications

Games

Groups

Marketplace (1)

Birthday Alerts

[More](#)[Chat with Friends](#)[Online](#)

News Feed

[Top News](#) · [Most Recent](#) **300+**

What's on your mind?



Alejandro Chavetta JFK > EZE JFK International Airport w/ <http://4sq.com/G6RHo>

[1 hour ago](#) · [Comment](#) · [Like](#)

Michelle Pae



Michelle just bought "The Fame" by Lady Gaga.

Buy it: \$9.99 at Amazon

See what else Michelle is buying.

[1](#) · [3 hours ago via BlueSpark](#) · [Comment](#) · [Like](#) · [See Michelle's Purchases](#)

Anne Paletti



Multi beach

9 new photos

[1](#) · [April 1, 2011](#) · [Comment](#) · [Like](#) · [Share](#)

Charlie Vestner



Charlie just bought an iPad!

7 of Charlie's friends also bought an iPad.

Do you have one too? Let interested friends know

Want one? [Add to Want List](#)

[1](#) · [1 day ago via BlueSpark](#) · [Comment](#) · [Like](#) · [See What Your Friends Are Buying](#)

Paul and a colleague placed an ad on Craigslist offering to pay people \$30 to come to their office and participate in social media market research. When people arrived, they were escorted to the research computer running the Greasemonkey script, asked to log in to Facebook, and then were asked to go through their newsfeed and comment on what they thought about Facebook.

Of the fifty people who went through this exercise, forty-seven hated all of the product data they saw. Said one test subject, “Facebook used to be a place where we shared our stories and pictures of our friends and family, but now it’s all commercialized!”

Within six weeks of being funded, Paul returned to his VC with both bad news and good news. The bad news was that the idea for which they funded his startup wasn’t going to work. The good news was that he hadn’t wasted their \$2 million investment. Paul had avoided falling into what my friend Melissa Perri calls the *build trap*:¹ building a product before validating whether it’s viable, feasible, and desirable.

The VCs were happy for Paul to continue experimenting, and he quickly and cheaply rejected six more ideas. His seventh idea, however, showed real promise. Paul hired engineers and started building a real service, called NeedFeed. After a little over a year, NeedFeed had more than five hundred thousand registered users. Unfortunately, NeedFeed failed when Facebook made changes to its site that severely limited what the fledgling service could do.

My friend William Pietri worked closely with Paul during this time. He said it was one of the best experiences of his entire career: “Creating social products is a high-risk/high-gain endeavor. I feel like we used our time and our investors’ money wisely.”

Agility in product development requires failing smarter—rapidly learning and adapting without spending lots of time or money. Doing that well is tricky, and it’s easy to make mistakes, like reaching false conclusions too hastily or crossing ethical boundaries. Laura Klein, a lean expert, user-experience guru, and author of *Build Better Products*, had the following to say about Paul’s testing:

If the test subjects weren’t seeing shopping data previously in their Facebook feeds and then they suddenly saw a lot of shopping data or purchases that seemed unappealing to them, that might automatically trigger the “this is new and I don’t like it” response. It’s often hard to distinguish between change aversion and something that users will actually dislike in the real world. It’s even harder to distinguish between the two using qualitative testing, because in research sessions people feel encouraged to comment on things they see that are different and then come

*up with reasons to like the changes or not like the changes that don't have anything to do with how they'd react under other circumstances. On the other hand, a strong, negative, visceral reaction to something in a user research session can be a useful signal, and it's certainly something I'd want to investigate further.*²

Testing ideas is indeed a tricky art, yet one that's well worth learning. Exactly how do you do that? David Bland and Alexander Osterwalder coauthored *Testing Business Ideas: A Field Guide for Rapid Experimentation*. It features forty-four fast and capital-efficient ways to test your ideas, and the book is well worth your time.

Leverage quick, cheap ways to test your ideas. You'll fail smarter, learn faster, pivot quicker, and increase your chances for success.

¹. Melissa Perri, *Escaping the Build Trap: How Effective Product Management Creates Real Value*, (Newton, MA: O'Reilly Media, Inc., 2018).

². From an email exchange with Laura Klein.



How Jeff Bezos Treats Failure and Success

The Fire Phone, Amazon's attempt at building and selling its own smartphone, was a total flop. The phone was released with much fanfare in July 2014, and after sales dropped precipitously over the ensuing months, Amazon discontinued the Fire Phone in August 2015. A reporter asked Amazon CEO Jeff Bezos to explain what happened. Without missing a beat, Jeff replied, "We're working on much bigger failures right now."

Jeff runs Amazon like a venture capital fund. He invests in many ideas and only expects a few to be smash hits. This works well because the hits easily pay for all of the failures.

In a 2016 letter to Amazon shareholders, Jeff Bezos wrote:

One area where I think we are especially distinctive is failure. I believe we are the best place in the world to fail (we have plenty of practice!), and failure and invention are inseparable twins. To invent you have to experiment, and if you know in advance that it's going to work, it's not an experiment. Most large organizations embrace the idea of invention, but are not willing to suffer the string of failed experiments necessary to get there.¹

Powerful words from a company that got to \$100 billion in annual sales faster than any other company in history.

In *Business Insider*, Bezos gave another wonderful insight about mistakes. He said, "The size of your mistakes needs to grow along with the company. If it doesn't, you're not going to be inventing at a scale that can actually move the needle."

Businesses that are afraid to make mistakes are brittle and subject to stagnation and disruption. Businesses that engender a culture where it is truly safe to fail simultaneously establish the conditions for a learning culture to establish itself—and are therefore better poised to adapt.

Size your mistakes to the size of your organization.

¹. Jeff Bezos to Amazon shareowners, 2016, sec.gov/Archives/edgar/data/1018724/000119312516530910/d168744dex991.htm.



Innovating Without Permission at Amazon

When Greg Linden was a summer intern at Amazon, he realized that the popular website wasn't yet offering personalized impulse buys. An impulse buy is an unplanned buying decision, usually made just before checkout. Chocolate bars, gum, and tabloid magazines are typical impulse buys in grocery store checkout lines, purposely arranged at eye level to attract your attention. Greg realized that Amazon could take this a step further by suggesting *personalized* impulse buys based on the contents of a user's shopping cart.

He was so excited about the idea that he quickly hacked together a prototype, installed it on an internal test website, and demonstrated the feature to others. Feedback was positive, except for one senior vice president (SVP) of marketing, who was not a fan. The SVP believed it was unwise to distract people during the all-important checkout process—potentially causing them to abandon the purchase altogether. So the SVP told Greg that he was forbidden from working on the feature.

But Greg refused to give up. In his account of the story, he said, “I believed in shopping cart recommendations. I wanted to measure the sales impact.” He prepared a version of his feature for an online test, which would allow him to gather data from a small fraction of actual Amazon users.

When the SVP heard about the online test, he was furious, but he was unable to block it. And then the results came in. Greg recalls, “Not only did it win, but the feature won by such a wide margin that not having it live was costing Amazon a noticeable chunk of change.” Amazon quickly launched Greg's feature. Personalized impulse buys at checkout would improve Amazon's revenue by 3 percent, resulting in hundreds of millions of dollars' worth of additional sales.

In some organizations, Greg would have been fired or penalized for disobeying an SVP. Hierarchy and the power to kill ideas merely based on opinion can severely limit innovation and revenue. Fortunately for Greg, even summer interns have the freedom to experiment at Amazon. Greg said, “Everyone must be able to experiment, learn, and iterate. Position, obedience, and tradition should hold no power. For innovation to flourish, measurement must rule.”¹

Organizations that empower people to test ideas rather than merely obeying a HiPPO (Highest Paid Person's Opinion) are more poised to innovate and adapt.

Don't let seniority, obedience, or tradition impede innovation.
Empower people to safely experiment and measure the impact of
their ideas.

¹. Greg Linden, "Early Amazon: Shopping Cart Recommendations," *Geeking with Greg* (blog), April 25, 2006, glinden.blogspot.com/2006/04/early-amazon-shopping-cart.html.



Product Ownership Is a Team Sport

A team of developers at a Fortune 50 automotive company was creating software for car dealers to manage vehicle orders. They'd been asked to build a feature that had a complex and awkward workflow, and the team had a better idea. It would allow the car dealers to access data quickly and easily via a spreadsheet and take a fraction of the time to implement. They presented their idea to the product owner (PO), who promptly rejected it without explanation.

Feeling perplexed and frustrated, the developers decided to build the spreadsheet feature anyway. They configured analytics to make sure they could report usage of the feature once it was released, and they did one more thing: the team kept its work secret from the PO.

Once the feature was live, the developers told the users about it. The users loved it, and the usage metrics proved it. The developers then showed the high usage data to the PO. The PO was not happy, and he was quite vocal about his unhappiness. His team had defied him!

The feature had taken less than four hours to implement and release to the customer and was an immediate hit. But this victory caused the team's relationship with the PO to sour. In fact, the relationship had already been suffering because the PO wanted total control over what to build. He felt he already knew how the application needed to be designed, including what it should look like and how it should interface with the customer. Features didn't need to be validated with customers since the PO was convinced that he already knew exactly what was needed.

Meanwhile, the software team included brilliant *craftspeople*—people who spent a great deal of time thinking about their users, connecting with them multiple times a week, asking them questions, observing their behavior, and listening to their most painful issues. These technical people had perfectly good ideas about what their users needed, but the PO was unwilling to hear them. He lived in a world in which certain people specify what to build, while others build it. He simply wanted his team to be busy building the features *he* specified—maintaining clear roles and responsibilities.

The developers, on the other hand, only wanted to solve their users' greatest pain points and build the most valuable software for them as safely and efficiently as possible.

Whose side would *you* take?

Unfortunately, too many people, including executives, product managers, and POs, fall into the trap of thinking that they alone should control what to build and be the sole decision maker. This approach limits a team's ability to be poised to adapt, diminishes what the team can achieve, and doesn't serve the customers of the product. In addition, talented members of the team will seek out other career opportunities where they are given more autonomy to implement their good ideas. When executives, product managers, and POs are truly open-minded to diverse ideas, willing to try experiments, and avoid the trap of sticking to traditional roles and responsibilities, they can collaborate with their teams to produce excellent results.

As Jeff Patton, an expert in agile product management and author of *User Story Mapping*, once said, "Product ownership is a team sport." You'll increase your chances for success dramatically if you allow team members to suggest and experiment with ideas and take an active role in producing happy customers.

Make product ownership a team sport.



Safety and Skill in a Rapid Pivot

One week before going into production with a web-based system created by a fledgling dot-com in San Francisco, a manager discovered that the price of a production license for a piece of middleware—software that provides services to applications—was extremely expensive. The company had been using a development license of the middleware product, which was inexpensive, and they were shocked when they learned how much a production license would cost. They called several developers into the boardroom and asked if anything could be done.

I was one of those developers. I had no particular affection for this middleware product. Before I arrived on the team, it had been chosen as a key component of the system because it could handle many difficult tasks. As the months passed, it became apparent that the middleware product wasn't actually all that great at the many tasks for which it was chosen. Eventually, the team used it for only one thing: object-relational mapping (ORM), a way for the software to get data in and out of a database.

We debated if we should persevere with this middleware or remove it altogether, thereby saving lots of money when the system went into production. The trouble was, we were just days away from a production release.

Some on the team said it was too late to make such a big change. Yet a majority of us felt we could safely remove the middleware entirely by pivoting to use a different ORM tool or building our own. The majority vote won the day and we got to work.

Three days later, our system was running perfectly without the middleware software, and a few days after that we went into production without a problem.

What enabled such agility?

Safety and skill.

We had a huge safety net of automated tests, which gave us a great deal of confidence to make changes. If any change we made to the code caused one or more tests to fail, it signaled to us that the change was unsafe and we could quickly remove it before the change caused a defect.

We also had a talented and fearless team. As we dug into the problem, it turned out that our ORM needs were not very sophisticated. This meant that creating our own ORM solution wasn't all that difficult.

Rapidly pivoting from the expensive middleware software to our own solution a few days before going into production is one of my favorite memories of being poised to adapt. We saved the fledgling dot-com lots of money, we

simplified the software design, and we gained new confidence in our ability to adapt rapidly thanks to safety and skill.

Safety and skill can help you adapt rapidly, and adapting rapidly
can be good for business.

YouTube is one of the most visited websites in the world and the second-largest search engine. One day, I navigated to YouTube's home page and found the following text: "We're redesigning YouTube. Please click here to experience the new look and feel." I clicked and found myself in the new design. It was cool and yet clearly unfinished—a work in process.

They wanted to know what existing users of YouTube, like me, thought of the new design. There was an easy way to give them feedback, so I provided some. They also made it easy for me to navigate back to the normal YouTube.

Most organizations don't learn fast enough. In the worst cases, they design or redesign in the dark, not seeking public feedback until after the new design has shipped. That's when they end up discovering lots of problems.

Smarter organizations learn faster. They have a habit of conducting safe, capital-efficient experiments that help them evolve excellent solutions. They don't hide things from their clients/users until those things are done.

Steve Blank, a deeply experienced entrepreneur who authored *The Four Steps to the Epiphany* and coauthored *The Startup Owner's Manual*, recognized that "there are no facts inside the building." In other words, inside of your company, you can only guess what customers really want or need. Steve exhorts us to "get out of the building" (GOOB) to learn from people, such as potential or existing customers.

GOOBing is integral to what Steve calls *customer development*, a practice that must coexist with product development. Get your ideas in front of people who don't work in your organization (and are thus outside of your building) and learn rapidly from them.

YouTube clearly understands this wisdom. They actively demonstrate how to learn faster by not waiting for work to be fully complete before seeking actual user experiences. This enables them to be poised to adapt.

Don't wait until something is finished before seeking feedback.

Get out of the building.



Are They Using It?

If you've ever visited a company that stocks snacks for employees, you've surely seen a refrigerator filled with drinks. Google is famous for having all kinds of snacks and drinks for its employees. But they go a step further than just randomly stocking their pantries. Google runs analytics on what people consume so they can do a better job of stocking the foods and drinks that people like.

Google is one of the most quantitative companies in the world. I had the pleasure of providing training and coaching services to their employees from 2004–2012. During that time, my colleagues and I developed an eLearning product that Google bought and deployed to thousands of students across their offices worldwide.

We made sure to provide Google students with the best possible learning experience. We made content improvements regularly, quickly answered student questions, and worked hard to provide an exceptional learning experience.

It all seemed to be going quite well. Until, a few years after we started, Google ran out of licenses and needed to buy more. That's when they wanted to look at some analytics. Specifically, they wanted to know how much their students were actually using the eLearning licenses they had acquired.

This was not data we'd been looking at. So we dug into the data and were totally alarmed at what we found. While the students had steadily consumed all of the licenses, the data showed that once a student acquired a license, they used only about 10–20 percent of it. Much of our eLearning had become “shelfware” for these students—completely overlooked, like a box on a shelf.

This usage data helped Google figure out that they should buy only the small fraction of content actually used by their employees. This led to a much smaller purchase of our eLearning.

Damn! We'd missed something huge and obvious! In hindsight, we were crazy not to actively monitor usage of our product for a client that is one of the world's most analytic companies!

This taught me that while you can strive to make an excellent product or service, if you don't pay attention to how people use your offering, you won't know how you may be failing your customer. And if you're failing your customer, it won't be long before this impacts your future sales. To be poised to adapt, you must know what your customers are doing or not doing with your product or service.

Actively study usage of your product or service to rapidly learn
how or what to adapt.



Usability Evaluations for Rapid Learning

I'll never forget the shock I felt one day while observing someone using a product I had helped build. The product, an eLearning system to help programmers learn agile software development, was already a few years old. We wanted to learn what most needed to be improved. My friend, and usability expert, Grant Young agreed to help us by conducting a usability evaluation.

Grant and I identified a sequence of scenarios—typical actions that users needed to perform while using our product. We then recruited programmers to participate in the usability evaluation. Grant would meet the programmers at our office and guide them through the scenarios while we filmed and observed them. Grant allowed me to observe these sessions, provided that I remained silent.

A fellow named Adam was our first participant. After welcoming him to the study, Grant guided Adam to perform one scenario after another. I watched from a sofa behind Adam and Grant, having agreed with Grant not to utter a word.

During one of the scenarios, Adam came to a page in our eLearning system that featured an instructional video. These videos were about three minutes long, yet they were a critical part of our eLearning courses. It could take us several days to produce a good three-minute video.



I watched as Adam arrived on the page with the video embedded in it. I watched as he appeared to look at the contents of that page. I expected him to click on the little triangle in the bottom left corner of the video to start watching it.

But Adam didn't click on the little triangle in the bottom left! Instead, he clicked the "Next Page" button to go to the next page. He'd completely missed playing the instructional video that we had labored over for days!!



I was *shocked* and it was hard to remain silent. Had Adam not seen the little “play” icon to start the video? Wasn’t it obvious? How could he completely miss it?! I wanted to leap up and show him what he had missed and how to play the video, but I had sworn not to say a word.

That experience is forever etched in my mind. We immediately fixed the issue, producing large, hard-to-miss splash screens for each video, featuring phrases like “Press to Play Video.” We retested and everyone clicked on the video, just as we intended them to.

We learned a lot from Adam that day, and from the others who participated in the usability evaluations. We had been humbled, and it helped us produce a far better product.

Be sure to study people using the things you make. You may be surprised (and humbled) by what you learn.



The Day We Stopped Sprinting

I adopted sprints (short, fixed-length time boxes, approximately one or two weeks in duration) in the late 1990s and used them for over seven years in software development. One day, in 2007, my colleagues at Industrial Logic and I decided to try something different. We'd recently learned about Kanban, a method inspired by lean manufacturing that didn't require sprints. While we didn't know much about it, we decided to try our own hand at working without sprints.

We would gather together to select a small, valuable piece of work to do, complete the work, deliver it to production (the computers used by our customers), and repeat. We still used rough, time-based estimates ("That work will take around two days," or "That work will take around two weeks, so how can we make it smaller?"). And we regularly performed bargain hunting—a practice for considering numerous options for any given piece of work that helped us discover "high value at low cost."

Our new approach felt more natural and less heavy. We spent less time predicting, managing, and accounting for how much work could fit into a sprint and more time focused on identifying and delivering value to our customers. We settled into a cadence of delivering new functionality within a few hours or days.

This new cadence required us to talk more regularly about what was most important to do. That led to short, focused huddles in which we continually challenged ourselves to identify our most valuable work. And that led to constantly thinking about what would make our customers happier and more successful or what we needed to improve internally (e.g., improving the design of our software).

Shifting away from sprints made us faster, more reflective, and more poised to adapt. I'm often asked if this style of working is only good for experts. The answer is no. In fact, based on experiences with many teams, I've found that it's harder for beginners to succeed with sprints than without them.

Shifting away from sprints was a milestone in my agile journey. It was a joy to discover how something I once thought was vital to agility (sprints) was not vital at all, how dropping sprints helped make software development faster, easier, and more graceful, and how my team could become even more poised to adapt to customer needs. This experience of dropping sprints paved the way for many more process improvement experiments.

Become more poised to adapt by regularly experimenting with
better ways of working.



Summary: Be Poised to Adapt

Agility requires being poised to adapt, and that takes practice. Commit to a daily pursuit of excellence. Practice how you respond to problems. Prepare for the unexpected.

You will have happier customers if you can rapidly adapt your products/services to their exact needs. Focus on a clear value proposition and then use multiplicity and selection to learn and adapt rapidly.

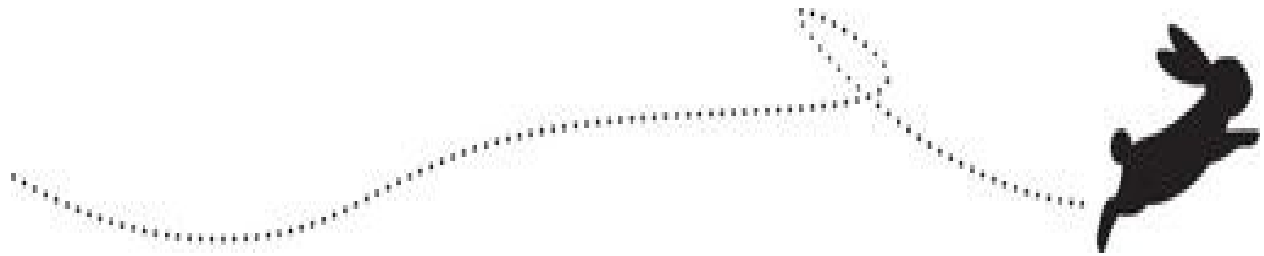
Find ways to fail better, faster, and cheaper in order to succeed sooner. Being poised to adapt is easier with a balanced team engaged in iterative experimentation.

Leverage quick, cheap ways to test your ideas so you can learn faster and pivot quicker. Size your mistakes to the size of your organization (e.g., Amazon's Fire Phone).

Don't let seniority, obedience, or tradition impede innovation. Empower people to safely experiment and measure the impact of their ideas. Get fast, frugal feedback so you can avoid sinking time and money into things your customers don't want or need. Make product ownership a team sport.

Safety and skill can help you adapt rapidly, and adapting rapidly can be good for business. Don't wait until something is finished before seeking feedback. Get out of the building to learn from actual customers. Be sure to study people using the things you make. You may be surprised (and humbled) by what you learn.

Become more poised to adapt by regularly experimenting with better ways of working.



Start Minimal and Evolve

When you don't know exactly what to create, what will delight customers, or what will help you achieve your goals, it helps to start with something minimal and iteratively improve it. Instead of guessing at what to create, acknowledge that you are operating under conditions of uncertainty and use an approach that helps you quickly and easily learn what to create. That helps you Drive Out Fear, a closely related mantra. Start Minimal and Evolve is also closely related to the mantra Be Poised to Adapt, since it requires rapidly adapting as you learn what to evolve.

When artists and architects progress from early sketches to a finished work, this mantra is at work. When startups begin with a rough, crude version of an offering so they may learn and pivot quickly without investing lots of capital, this mantra is at work. And when a seed begins life as a tiny seedling and eventually grows into a giant tree, this mantra is at work. There is no better way to be quick, resourceful, and adaptable than to start minimal and evolve.

Evolutionary design enables this mantra. It prompts us to begin with a primitive whole, learn from it, evolve it, and repeat.



Cheap, Minimal Teleportation

“Why can’t my avatar move around this environment?!” That was the consistent feedback from early customers of IMVU, a virtual world startup that provided an avatar-based social experience. Users of *The Sims*, the industry’s leading virtual environment, could freely move their avatar through the environment, entering buildings, walking around, and avoiding obstacles. Why couldn’t IMVU users do that?

IMVU was cofounded in 2004 by Eric Ries, an entrepreneur best known for his book *The Lean Startup*. Prior to IMVU, Ries had worked at There.com, another virtual world startup, which failed miserably after spending millions of dollars over several years building software that no one ended up wanting to use.

Ries was determined to make IMVU succeed. To do that, he leveraged customer development wisdom from serial entrepreneur and professor Steve Blank, principles from lean manufacturing, and practices from agile development. The process he developed was fast and frugal, minimizing large, speculative investments by first validating or invalidating ideas quickly and inexpensively.

Enabling avatars to move through the IMVU virtual world—the way they could in *The Sims*—would be expensive and time consuming. Several engineers estimated that it would require an entire software team working for at least three months.

That didn’t sound fast and frugal. So the team decided to first experiment with a much cheaper option. Ries described it in *The Lean Startup*:

*We used a simple hack, which felt almost like cheating. We changed the product so that customers could click where they wanted their avatar to go, and the avatar would teleport there instantly. No walking, no obstacle avoidance. The avatar disappeared and then reappeared an instant later in the new place.*¹

It took just two days to build and release the teleportation feature. The team figured that if no one liked it, they could easily remove it and then invest the time and money to build a more sophisticated movement feature.

But then something unexpected happened: positive customer feedback! IMVU regularly polled users to ask them about their top-three favorite features of the service. Shockingly, the quick, cheap teleportation feature started showing up regularly as a favorite feature. One person even said, “I like this even better than *The Sims*! I can just click and be somewhere instantly!”

Learning rapidly how to delight customers via capital-efficient

experimentation became a hallmark of IMVU. The cheap, minimal teleportation hack helped this fast and frugal startup solve a problem with less code, less work, less complexity, less time, and less money.

When is less more? Start minimal and learn.

[1.](#) Eric Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (New York: Crown Business, 2011).



Deliver Value Sooner

American Airlines filed for bankruptcy protection in 2011. The company's profitability problems resulted from numerous factors, including the high cost of fuel, labor, and pensions. During this difficult time, employees felt the pressure to find new ways to generate revenue.

Some product managers realized that the last-minute sale of first-class seats was an untapped revenue source. When economy passengers checked into their flights on the company's website (AA.com), American Airlines could offer them an upgrade to first class at bargain-basement prices. The company would fill its first-class seats while earning extra revenue in the process. The product managers asked development how long it would take to produce this functionality.

While it sounded simple, the details were nontrivial. The upgrade-at-check-in feature would need to handle many scenarios, such as ticketing a new reservation and modifying an old reservation when four passengers shared one reservation, but only two of them wanted to upgrade; or when passengers had multiple legs on their journey, but only wanted to upgrade one leg. You can imagine all of the possibilities. The product managers also wanted passengers to be able to upgrade using any form of payment, including promotion codes, gift cards, vouchers, and credit cards.

Mike Rieser, a project lead and agile coach, along with a small team of colleagues at American Airlines, studied the software to formulate an estimate. They quickly realized that it would take several months to program everything requested by the product managers. They did not want to be the bottleneck, especially at a time when the company badly needed to increase revenue.

So, rather than giving an estimate to program all requested functionality, they suggested a different idea: initially ship a limited version of the upgrade feature that only supported a single form of payment (credit cards) for passengers traveling alone on single-leg trips. After releasing that version, the team would roll out successive versions, until every upgrade scenario and payment option was handled.

This approach is called *evolutionary design and delivery*. You start with something minimal and evolve it over time. American Airlines could use this approach to begin collecting new revenue sooner, rather than waiting for a "big bang" release of the fully functional, first-class upgrade feature. Think of it as a "little bang."

Now it was up to product managers to decide if they could go along with this

evolutionary plan. If they said no, then *they* would be the bottleneck. They decided to accept the plan.

The simplified upgrade capability was soon live on [AA.com](#), generating much-needed revenue. Over the succeeding days and weeks, Mike and the team pushed more deliveries, adding support for complex upgrade scenarios. Before long, the website could handle all of the scenarios and diverse payment options.

Mike and the team had found a way to deliver value sooner. You can achieve this by understanding and practicing evolutionary design and delivery.

Within every sophisticated solution lies a more basic version waiting to be delivered sooner. Instead of delivering a big-bang release, begin with a little bang.



An Evolutionary Path

In the early 1970s, a new building on the University of Oregon campus was finished just in time for the start of a new semester. Only one detail remained: constructing paved pathways to the building. When classes commenced, students had to walk to the new building across grass and dirt.

And that was precisely the way Christopher Alexander, author, professor, and lead architect for the building, wanted it. Rather than guess where to place the paved pathways, Alexander and his team decided to wait and see where people naturally walked.



It didn't take long to find out. Foot traffic to the new building gradually revealed common pathways in the dirt and grass. Once these preferred student and faculty paths became visible, Alexander and team knew exactly where to

construct the permanent paved pathways.

In the book *The Oregon Experiment*, Alexander and his coauthors called this the *principle of organic order*: planning and construction will be guided by a process that allows the whole to emerge gradually from local acts.

Evolutionary design follows this principle: observe and learn how people use a minimal solution in order to know how to improve that solution.

Artists, filmmakers, software designers, writers, and many others also work this way. Sketch, craft, refine. Dirt to pavement. Start minimal and evolve based on insights and learning.

Had Alexander and team simply guessed where to place the paved pathways, the students and faculty might have bypassed the pavement in favor of more expedient paths. Instead, the architects had the humility to know what they didn't know and a quick way to learn what they needed to know.

Start with something basic, then observe, learn, and evolve.
Evolutionary design helps you discover what you *really* need to
build.



Begin with a Primitive Whole

To evolve great products or services, it helps to practice evolutionary design. In order to do that, you need to understand what I call a *primitive whole*.

Like an artist's sketch, a primitive whole is a fully assembled, yet extremely underdeveloped version of a finished whole. The main parts are all there, but they are primitive. The primitive whole doesn't do everything that the finished whole will eventually do. While you may envision a beautiful and sophisticated product or feature, you begin with a basic, immature implementation.

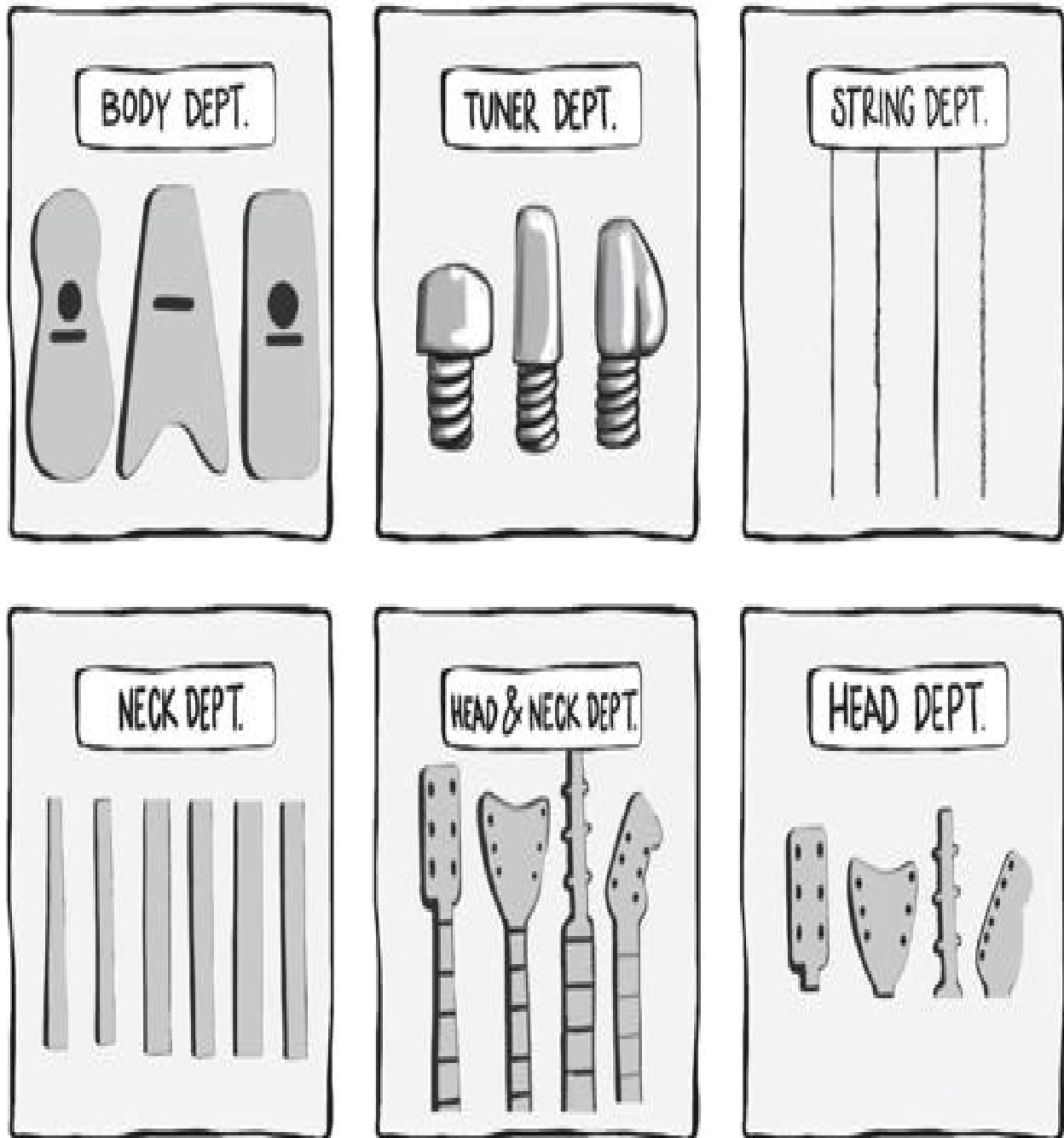
Consider how the guitar below evolves from a primitive whole to a finished guitar:



Wait a minute, is that first guitar even a guitar?!? It only has one string, no

tuning pegs, no frets, saddle, or bridge and even lacks a sound hole! And yet you can play entire songs with it, including “Thunderstruck” by AC/DC, “Sunshine of Your Love” by Cream, and “Seven Nation Army” by The White Stripes.

That first guitar is a primitive starting place, an underdeveloped whole that will evolve based on needs, priorities, risks, and feedback. It’s very different from completing a bunch of unassembled guitar parts:



What’s so wrong about completing a bunch of parts instead of beginning with a primitive whole? Consider people on a team or spread across several

teams often working in silos, making progress on building “parts” of a solution without frequently communicating or integrating their work. Much later on, they may discover that the parts they built in isolation don’t integrate correctly with other people’s parts. Or worse, the final assembled solution doesn’t meet the customer’s needs because those customers were never involved in helping to shape and guide the work. Evolutionary design helps guard against such common problems by directing us to collaborate, integrate, and learn early and often.

And that takes skill.

Given a description of a fully functioning feature or product, you need to be able discern the primitive starting place. Start by saying “no” frequently at the beginning. For example, in the earliest version of a new payroll software system, we could have no accounting for sick leave, no support for vacations, and no handling of federal or state taxes or leaves of absence. What?!? No one would ever want or use such software! Correct. The primitive whole is a starting place, a foundation upon which we evolve more functionality or details.

Saying no early on does not mean you release something worthless. Apple’s first iPhone did not record video, did not have a copy/paste function, did not let you update the wallpaper, and did not do a host of things that modern phones do today. It didn’t matter. It was an excellent starting product. Evolutionary design is about gradually and gracefully evolving your offering, adding, changing, or removing functionality, and getting feedback along the way. You get to decide what is or isn’t releasable and how to improve your offering over time.

Start with a primitive whole by saying “no” to everything but the essence.



Birth of an Online Shoe Company

In 1998, Nick Swinmurn was in his early twenties, internet commerce was barely in its infancy, and entrepreneurs everywhere competed to bring anything sold in brick-and-mortar stores to the new online world. When Nick was unable to find a pair of Airwalk desert chukka boots at his local mall, he got the idea to start selling shoes online. With no knowledge of the footwear industry, no funding, no business plan, and no shoe inventory, Nick launched ShoeSite.com in early 1999.

The embryonic website, featuring pictures of shoes and a form for purchasing them, was created quickly and cheaply to validate whether people would actually buy shoes online. In an interview with *Business Insider*, he recounted, “I went to a couple of stores, took some pictures of the shoes, made a website, put them up, and told the shoe store, if I sell anything I’ll come here and pay full price. They said OK, knock yourself out.” Sales from his website quickly proved that people would buy shoes online. It was an early victory, executed with quick, easy grace.

Growing the business was a harder problem. Nick needed to raise capital. He approached venture capitalists (VCs), but the VCs didn’t believe that people would buy shoes online without first trying them on. At every VC meeting, Nick explained that one in twenty shoes were already being sold via mail-order catalogs to people who had never tried on the shoes before buying them. It made no difference. The VCs said, “Yeah, but people won’t buy shoes online,” or “It sounds like there is already a successful mail-order business.”

Nick eventually raised \$150,000 via assorted friends and even his chiropractor. This enabled ShoeSite to hire a few people and continue to grow.

Six months into the venture, Nick learned that a company named Venture Frogs might consider investing in his startup, so he called and left a voicemail on their answering machine. Twenty-four-year-old Tony Hsieh listened to it. He had recently sold his own startup, Link Exchange, to Microsoft and was now investing in other companies. Tony recounted,

I almost deleted the voicemail. Nick left a message saying he wanted to start a company that sold shoes online. I didn’t think consumers would buy shoes sight unseen, and Nick didn’t have a footwear background. It sounded like the poster child of bad Internet ideas. But right before I hit delete, Nick mentioned the size of the retail shoe market—\$40 billion. And the more interesting thing was that five percent was already being done through mail-order catalogs. That intrigued me. Initially, I was just an adviser. But I

*got sucked in.*¹

Tony invested \$500,000 in the startup, and together with Nick, they changed the company's name to Zappos, based on the word *zapatos*—Spanish for “shoes.” Nick and Tony hired a shoe veteran from Nordstrom who brought invaluable expertise to the business. In the years that followed, they survived the dot-bomb era and grew the business to become a darling of the internet. Zappos was ultimately acquired by Amazon in 2009 for \$1.2 billion.

Had Nick begun his venture by investing months of time building a full-featured website and purchasing a large inventory of shoes, ShoeSite might never have survived. Instead, Nick began with an idea, validated it without spending a lot of time or money, found investors, grew a team, and together with Tony, built one of the most successful internet commerce businesses of all time.

Knowing how to start minimally and evolve is easier with good, validated research and the right team.

¹. Tony Hsieh, “How I Did It: Tony Hsieh, CEO, [Zappos.com](https://www.zappos.com),” *Inc.*, March 31, 2022, inc.com/magazine/20060901/hidi-hsieh.html.



The Ballroom

I’ve got to bring this back to my company,” said Don Carlson, an executive at the TV ratings giant Nielsen Media Research.

Don had just attended a five-day Extreme Programming boot camp in San Francisco run by my colleague Somik Raha and me. Don loved what he experienced in the workshop, as it reminded him of how he had worked years earlier at Nielsen—before the culture changed. So he invited me to visit the company and see what we could do together.

When I arrived in sunny Oldsmar, Florida, I had the opportunity to meet Katherine Thalheimer, the director of one of Nielsen’s largest divisions. “All of our projects are completed on time and within budget,” Katherine told me. “And that’s the problem,” she continued. “Everyone builds a huge cushion into every initiative, so even tiny projects take way too long and cost too much.”

Don and Katherine wanted my company to help this Nielsen division become genuinely agile so they could adapt faster to market changes and deliver quality solutions more quickly. One such market change was a huge shift in TV viewing habits. An increasing number of people were recording shows to watch later rather than watching the shows live. Nielsen’s clients wanted the company to provide updated analytics on this “watch later” style of TV viewing, since it impacted multimillion-dollar advertising decisions.

To supply the updated analytics, Don had assembled a thirty-person balanced team consisting of mainframe, data warehouse, and Java programmers, testers, analysts, product managers, and advertising experts. This extraordinary group became one of the highest-performing teams with whom I’ve ever worked. They succeeded in helping Nielsen adjust to the tidal shift in television viewing habits.

Prior to starting work with the thirty-person team, my colleague Somik provided the Nielsen employees with a list of several books to study. One of them was Patrick Lencioni’s classic *The Five Dysfunctions of a Team*. At regular lunch-n-learns, team members would meet to discuss these books, and this was the beginning of the team’s journey to a culture of continuous learning.

With Don’s help, we secured an excellent physical team space. It was a large room consisting of “caves” (places to work privately or in small groups) and “commons” (the main work area, configured to comfortably support collaboration). The collocation of this team was extremely helpful, especially because they were a large team. Over time, the room became known as “The Ballroom” because people had learned to regularly “pair up” (like dance couples) to collaborate on their work.

Next, the agile coaches and I spent a week training the team in agile management, planning, and development principles and practices. This involved simulations and games, as well as numerous half-day iterations to plan and build software and then reflect on how well we did the work. After the week of intensive training, everyone knew a lot more about the agile process we were helping them adopt.

Next, we began coaching the team on evolutionary design: the art of evolving a product from a minimal, early, internal version into a customer-ready shipping product. This team was assigned the task of transforming a giant (one-foot-thick) report used by advertisers to decide which advertisements to place in which TV shows. As a part of this transformation, the team had to create a new version of an older report that now had to handle a new data feed—as well as the older data feed—and update the math to handle both kinds of data.

We helped the team plan and produce a “walking skeleton”¹ of the report, and later add functionality by “adding flesh to the bones.” The first walking skeleton was a one-line version of the giant report. It was mathematically correct, used the old and new data feeds, and took less than two weeks to produce by the mainframe, data warehouse, and Java programmers.

Once the team had a one-line report, they moved on to the next scenario, and the next, and the next—gradually evolving the report to become a shippable product. The team picked up momentum, eventually needing to pull work from their “Release 2” into “Release 1” because they were so far ahead of schedule.

This team achieved stunning success for Nielsen. They shipped a critically important product in record time. And because of the great success of the project and agile implementation, the team was awarded the Chairman’s Award for Process Improvement by Nielsen’s global holding company. Of the company’s twenty thousand employees worldwide, hundreds of projects had been submitted for the award. Only ten were selected as winners, putting the team in a very elite group.

The Ballroom became a popular place for executives from other divisions and sibling companies to visit and observe a deeply agile team at work. Katherine and Don soon began wondering how to spread this success to other parts of their division.

Begin with one great success and evolve from there.

1. wiki.c2.com/?WalkingSkeleton.



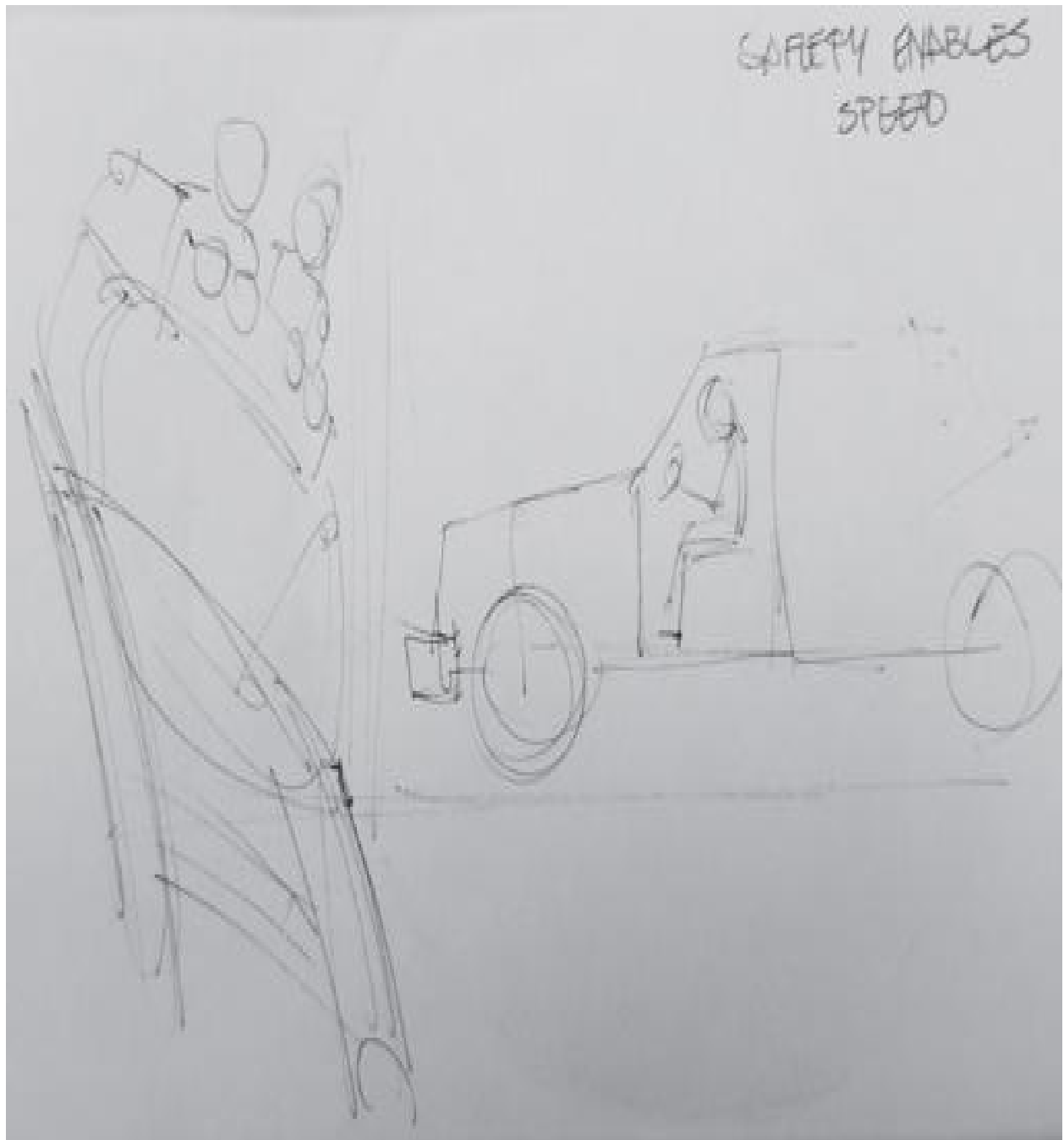
Sketch, Craft, Refine

A fine piece of art, be it a painting in a museum or an illustration in a book, begins life in a primitive form and evolves to become great. Study Picasso's famous paintings and you'll discover that he first began with a series of sketches, chose one to develop further, crafted a painting from the sketch, and ultimately refined the painting into a stunning work of art. "Sketch, craft, refine" is an abundantly agile process.

Over the years, I've worked with a half-dozen artists to produce illustrations for my business. I often don't know exactly what I want, especially when I'm trying to convey an abstract, intellectual concept. I begin by suggesting an idea, roughly conveying what I'm trying to communicate. The illustrator quickly produces sketches, using the crudest of tools and not fussing around with slow, complicated details. Every sketch lives a short life, helping us learn and adapt rapidly, without investing a lot of time and energy.

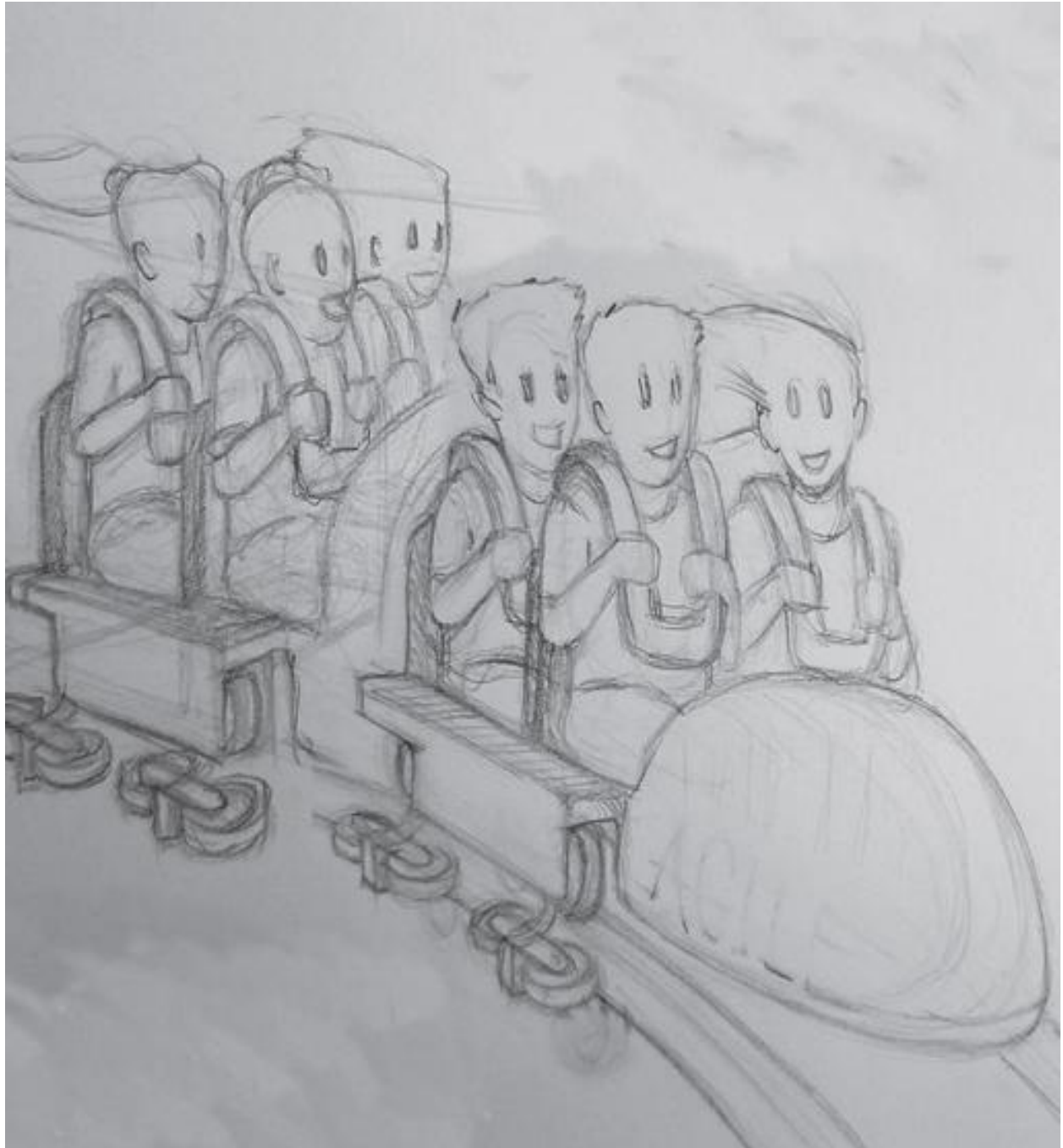
Sometimes none of these early sketches are right and we need a new batch. Other times, one of the sketches is worth developing. These fast, cheap sketches help us rapidly decide what the illustration needs to become. Sketching is perhaps the most important part of producing a great illustration.

Before I hired Julius Jervoso (the talented artist who produced most of the illustrations in this book), I asked him, during his first interview, to sketch whatever came to mind to illustrate the idea "safety enables speed." Without hesitation, Julius used a pencil to sketch two ideas on a single index card:

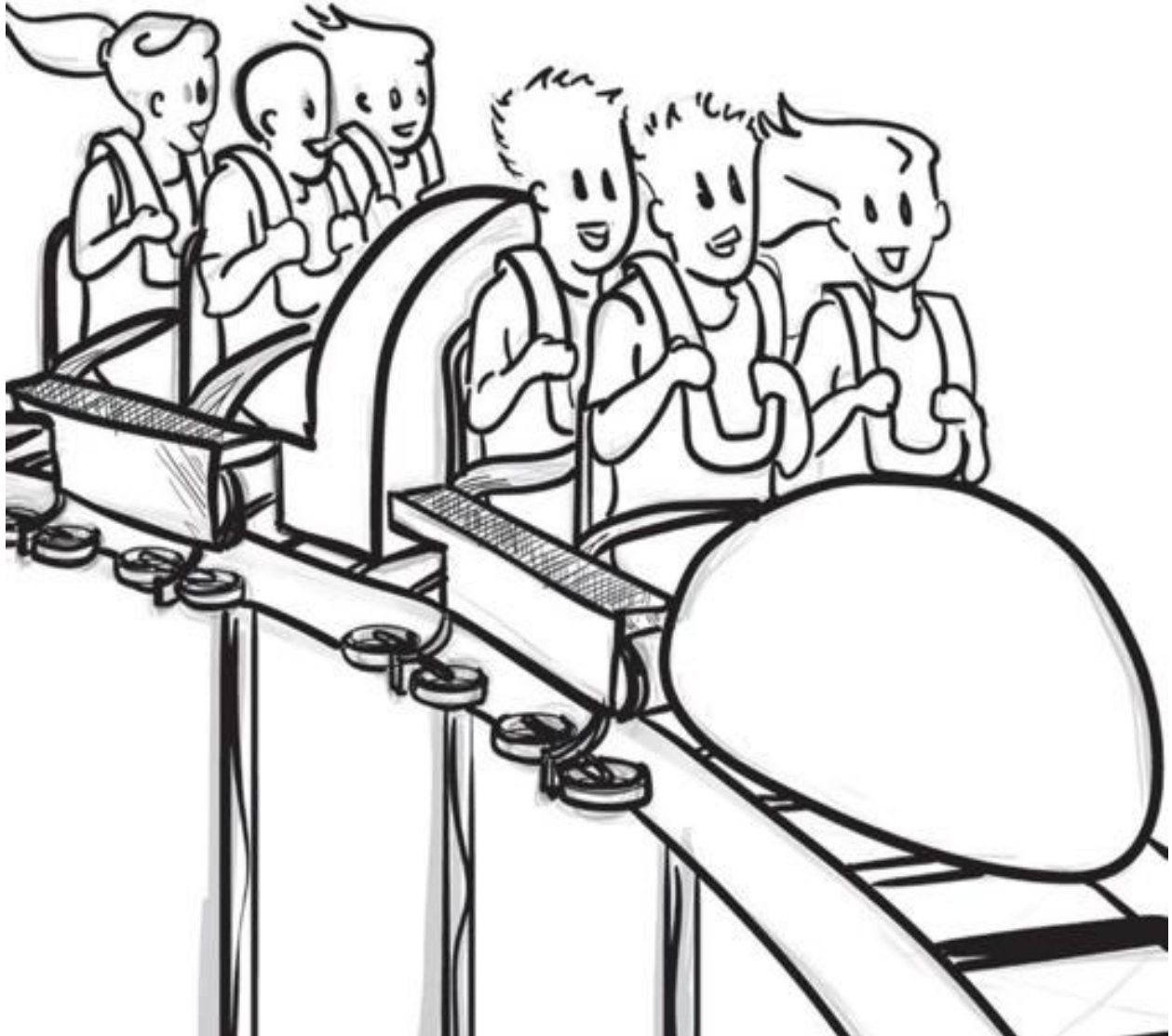


I loved the sketch and the idea of people safely riding a high-speed roller coaster. After a few more rounds of interviews, my company hired Julius.

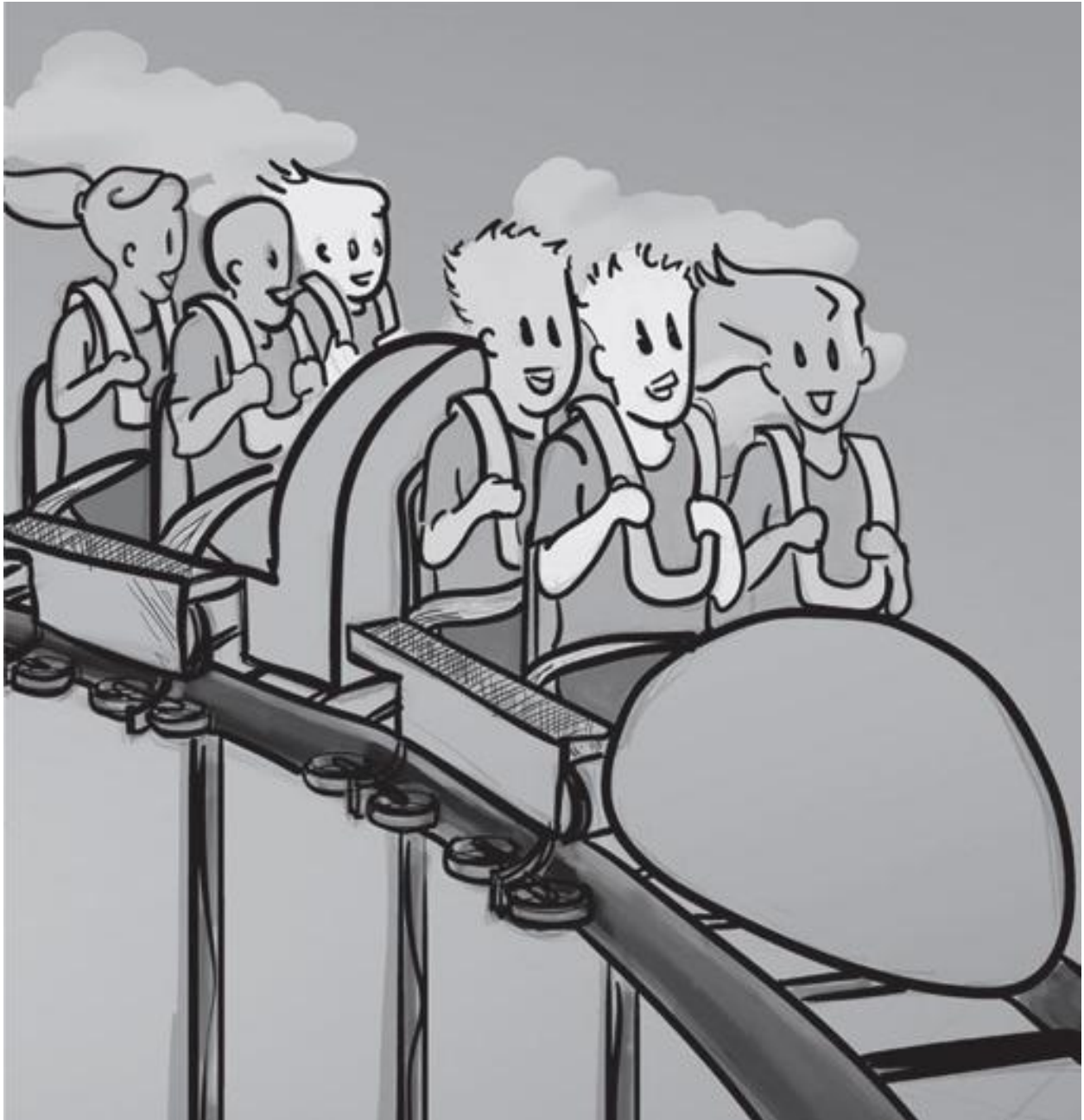
A few weeks later, I asked Julius to produce an illustration for “safety enables speed” that I could use in presentations. We discussed some ideas to depict safety on the roller coaster and Julius crafted the following illustration:



I was delighted with this work. It contained far more detail than the initial sketch, and it took longer to complete, yet all of that made sense, since we had decided to invest in this idea. Julius and I shared the sketch with others, got feedback, and then Julius continued to craft the illustration, this time using digital tools:



Refinement, the final step in the process, involves making any improvements necessary to finish the piece. While adding color often happens during the crafting stage, Julius handled that in the refinement stage, to produce the following finished illustration:



“Sketch, craft, refine” is an agile process that isn’t just for artists. I’ve used it to evolve proposals for clients, redesign the layout of a room, build a feature in a software product, and more. It’s an invaluable method for growing something great.

Sketch, craft, and refine is an agile way to evolve something great.



Evolving a New Ticketing System

What does evolutionary design have to do with buying train tickets in Norway?

Thorbjørn Sigberg, a Norwegian agile coach, told me this story, which beautifully illustrates how evolutionary design helped Norway's railroad company quickly and easily upgrade its outdated ticket purchasing system.

The old system ran on hundreds of computers located in train stations and convenience stores throughout Norway. It was built before browsers and web technologies existed. Making changes to it required manually updating the software running on each and every machine, one by one.

Modernizing to a web-based solution would allow the railroad to quickly and easily release improvements. Thorbjørn and a product manager met over drinks to consider how best to approach this upgrade. The existing software allowed people to buy one-way and round-trip tickets and weekly and monthly passes, specify which train stations to travel between, and request ticket refunds.

Thorbjørn and the product manager quickly rejected the idea of spending many months producing the complete web-based software necessary to handle every possible aspect of buying train tickets and integrating with all the various systems (the host machine in the train station/convenience store, the payment processing service, and the main back-end railroad system).

Instead, they decided to evolve a solution.

The initial system they envisioned would only allow people to buy a single, one-way ticket from the train station next to one particular convenience store, going to the train station in the next major city. There would be no option to select what station to start from or go to. You wouldn't be able to buy a weekly or monthly pass. And there would be no refund capability to start. This minimal pilot would help the development team drive out risks by learning what was necessary to make the software communicate with the store's register, the payment service, and the back-end railroad system.

A few weeks after hatching this plan, the first web-based pilot system went live in one local convenience store. It had only one button. If you clicked it, the register in the convenience store was notified, a payment system processed payment, and an order was issued to the railroad for a single ticket from the train station next to that convenience store going to the next major city. Soon after going live, actual customers walked into the convenience store and bought tickets!

After this successful pilot, the team increased to two convenience stores.

Doing that immediately exposed a defect related to authentication. After the defect was fixed, they added more functionality to the system and increased to five pilots. After everything was verified to be working correctly, they activated the entire system throughout Norway. With that completed, they could then realize the objective of being able to quickly and easily release improvements to the ticket purchasing software in all train stations and convenience stores.

Start minimal, experiment, evolve, and deliver value with quick, easy grace.



Birthing a Best Seller

By late 1999, I was ready to write a serious book. My friends in the software industry had already published some great books. It was my turn.

At the beginning of this arduous journey, I remembered these words from a prolific author and entrepreneur named Don Lancaster: “There is no point in writing a book if it’s not a runaway best seller.”

How might I guarantee that?

First of all, I knew I needed deep expertise in my subject matter. I’d been immersed in two areas of software design for many years: patterns and refactoring. I’d learned valuable lessons from which I thought a lot of people could benefit, and what better way to teach them than via a book? But how could I be sure?

I decided to start with an article. I called it “Stop Over-Engineering,” and it was published in *Software Development* magazine. I also repurposed it as a paper for a conference. Feedback was positive, so I decided to take the next step.

Around this time, a guy named Bruce Eckel published a book called *Thinking in Java*. What was remarkable about Bruce’s writing process was that every change he made to his rough, unfinished manuscript was published to the web. Anyone could download it, provide feedback, and follow along as he wrote his book. I decided to follow Bruce’s approach.

I began writing *Refactoring to Patterns* in early 2000. It included the “Stop Over-Engineering” article, and as I added more content, I published it to my website and let people know about every new version of the embryonic manuscript. I quickly started getting useful feedback, making sure to thank people for their thoughts and showing them updates I’d made based on their observations.

A lot of authors seek out a publisher before they write their actual manuscript. I didn’t do that. I was fortunate enough to have the advice of another best-selling author and guru named Gerald M. Weinberg. He’d written more than thirty books, most of which were best sellers. He told me to write my book on my own timeline and not get a publisher, as doing so would impose a deadline. That would in turn cause pressure that would interfere with me discovering what I wanted to say and what I needed to learn while writing my book.

So I wrote my book without any publishing contract. I also taught workshops on the topic, sharing the manuscript with students. These workshops helped me to test and refine my content. Meanwhile, more feedback rolled in from random strangers on the web. One person from HP sent me twenty-eight pages of

feedback. I was blown away by the quantity and detail of this feedback. Another individual, John Brant, an expert in refactoring and patterns, helped me understand that I had more to learn about the subject of my own book! His help was invaluable in producing high-quality content.

By late 2003, after a lot of hard work during nights and weekends (I still had a day job), I was getting close to completing my book. I decided to finally seek publishers, and to my surprise, three publishers engaged in a heated bidding war for it. Surreal doesn't begin to describe how the experience felt. Each publisher kept outbidding the others. I even got a call from a famous author, encouraging me to choose *his* publisher, which was part of the bidding war.

My book was published in 2004. The first printing was ten thousand copies, which made it an instant best seller for technical books. Tens of thousands of copies of the book have been published and sold, and it has been translated into ten languages.

My process for writing this book was steeped in continuous learning, nonstop refinement, and active marketing (a word I seldom used at the time). By speaking about *Refactoring to Patterns* at conferences, teaching workshops on it, publishing public versions of the manuscript, and alerting others about free online versions, I was in fact marketing the book. And that marketing, together with my expertise and a whole lot of learning and careful writing, helped me create a best seller. Don Lancaster would've been proud.

Give yourself the time and space to evolve something
exceptional.

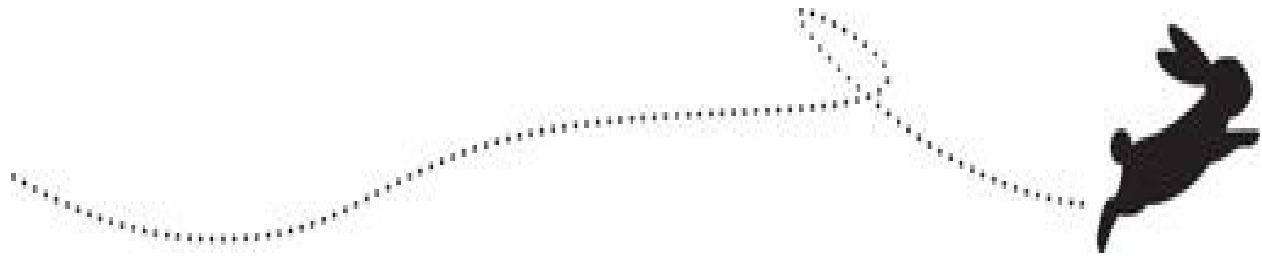


Summary: Start Minimal and Evolve

When is less more? Start minimal and learn. Within every sophisticated solution lies a more basic version waiting to be delivered sooner. Instead of delivering a big-bang release, begin with a little bang. Start with something basic, observe, learn, and evolve. Evolutionary design helps you discover what you *really* need to build.

To produce a primitive whole, say “no” to everything but the essence of what you ultimately plan to produce. Knowing how to start minimally and evolve is easier with good validated research and the right team. Begin with one success and evolve from there.

Sketch, craft, and refine is an agile process for evolving something great and delivering value with quick, easy grace. Don’t rush evolution. Give yourself the time and space to evolve something exceptional.



Drive Out Fear

When we are agile, we are unafraid. We don't hesitate or worry about what people think or what penalty might occur if we make a mistake. We are fearless.

Fear inhibits agility. It produces hesitation and delays. It leads people to remain silent over speaking up. It may encourage people to cover up mistakes, so they aren't blamed or penalized. Issues don't get resolved, which leads to compounding problems and failure.

Fear has no place in high-performance environments. That's why W. Edwards Deming famously said, "The manager's job is to drive out fear." In today's world, trust and safety are essential for high performance. And the research proves it. Paul J. Zak, a pioneer in neuroeconomic research, found that a high-trust environment produces chemicals in our brains that result in greater engagement and retention, lower stress, and more joy. Google's two-year study of teams to discover what led to high performance concluded that the number-one factor was psychological safety.

Great leaders drive out fear and establish safety. My friend Richard Sheridan, cofounder and CEO of Menlo Innovations and author of *Joy, Inc.*, explains how he does this:

I like to think of our approach at Menlo as a cultural HVAC system. We pump fear out of the room, filter out ambiguity, adjust the cultural temperature to the setting that makes the team comfortable, and then pump safety back in.

When we pump fear out of the room and give the team permission to make mistakes, the team starts to feel safe. If team members feel safe, they will begin to trust one another. If they trust one another, they will begin to collaborate and we see teamwork. When mistakes are made, the team owns up to the mistake because there is no fear of reprisal or penalty. No time or human energy is wasted by organizing a posse to nab the culprit.¹

We can say, over and over again, that "lowering work-in-process" or "removing silos and slow handoffs" will increase performance; yet in many contexts, it won't happen. Why? Because people don't feel safe to make such

changes. Perhaps they lack the political capital to suggest collapsing four siloed teams into one balanced team. Or perhaps they are afraid of upsetting other managers. Or maybe they feel that upper management will simply ignore them. In such cases, many will conclude that it's safer to not upset the status quo and just "do the job."

What most people need, more than agile or lean methods, is safety to experiment and pursue necessary and important improvements. And that requires driving out fear.

[1.](#) Richard Sheridan, *Joy, Inc.: How We Built a Workplace People Love* (New York: Portfolio Penguin, 2013).



Make Safety a Prerequisite

At 11 PM one night in 1987, the phone rang in the Connecticut home of Mr. Paul O'Neill, CEO of Alcoa (the Aluminum Company of America). On the other end of the line was Tony, a low-ranking production worker at Alcoa's sprawling aluminum plant in east Tennessee. What could have possibly spurred Tony to call the CEO of this Fortune 100 business at home so late at night?

In 1987, Alcoa was a one-hundred-year-old industrial giant, struggling with stiff competition, quality problems, outdated processes, and a near-total lack of innovation. Several CEOs had tried to turn the business around, but nothing had worked. The board finally asked Mr. O'Neill to step in.

Before accepting the CEO job, O'Neill did some research. He asked financial analysts in New York what they thought about Alcoa, he visited Alcoa's largest customers—including Boeing and Anheuser-Busch—to understand what they thought about Alcoa as a supplier, and he went to Alcoa plants to look around and talk with people. After three months, he concluded that safety was the key to rescuing Alcoa from its multi-decade slump.

With this conclusion in hand, O'Neill's first order of business was to travel to Alcoa plants in the U.S. and abroad, explaining to every employee and hourly worker that "safety is a prerequisite," that "no one should ever get injured at work," and that his goal was to have an injury-free workplace. He told supervisors and workers that if anyone identified a risk that could cause someone to get hurt, he wanted them to fix it. If that meant spending money, he would figure out how to pay for it.

And he did one more thing that was completely unexpected. O'Neill handed every worker his home phone number and asked them to call him if their supervisor didn't listen to him and put them in harm's way.

Tony had heard O'Neill's speech but wasn't sure if the man was serious. Now his supervisor was asking him and his colleagues to perform a dangerous job. Three conveyor belts transported large, six-hundred-pound ingots of aluminum from the casting pit to the rolling mill, where they were turned into sheets of aluminum used in beverage can production. When the middle conveyor belt broke, Tony's supervisor asked him and his colleagues to manually carry the heavy ingots from the first to the third conveyor belt.

When Mr. O'Neill answered the phone, Tony explained the situation: "It takes a lot of us to lift this ingot off the working conveyor, over the broken conveyor to the next working conveyor, and we're worried about hurting our backs. If we drop this on our feet, we'd crush our feet."¹ He wanted Mr. O'Neill

to know that, despite what he'd said in his speech a few weeks earlier, safety wasn't real at the plant in Tennessee.

Mr. O'Neill thanked Tony for calling and told him that he and his colleagues did not have to perform the dangerous work. Then he called the plant manager to ask that the broken conveyor be fixed immediately and to call him back when it was working. At 4 ^{AM}, the plant manager called to say that the conveyor belt was fixed. By morning, word spread all across Alcoa that the new CEO really was serious about worker safety.

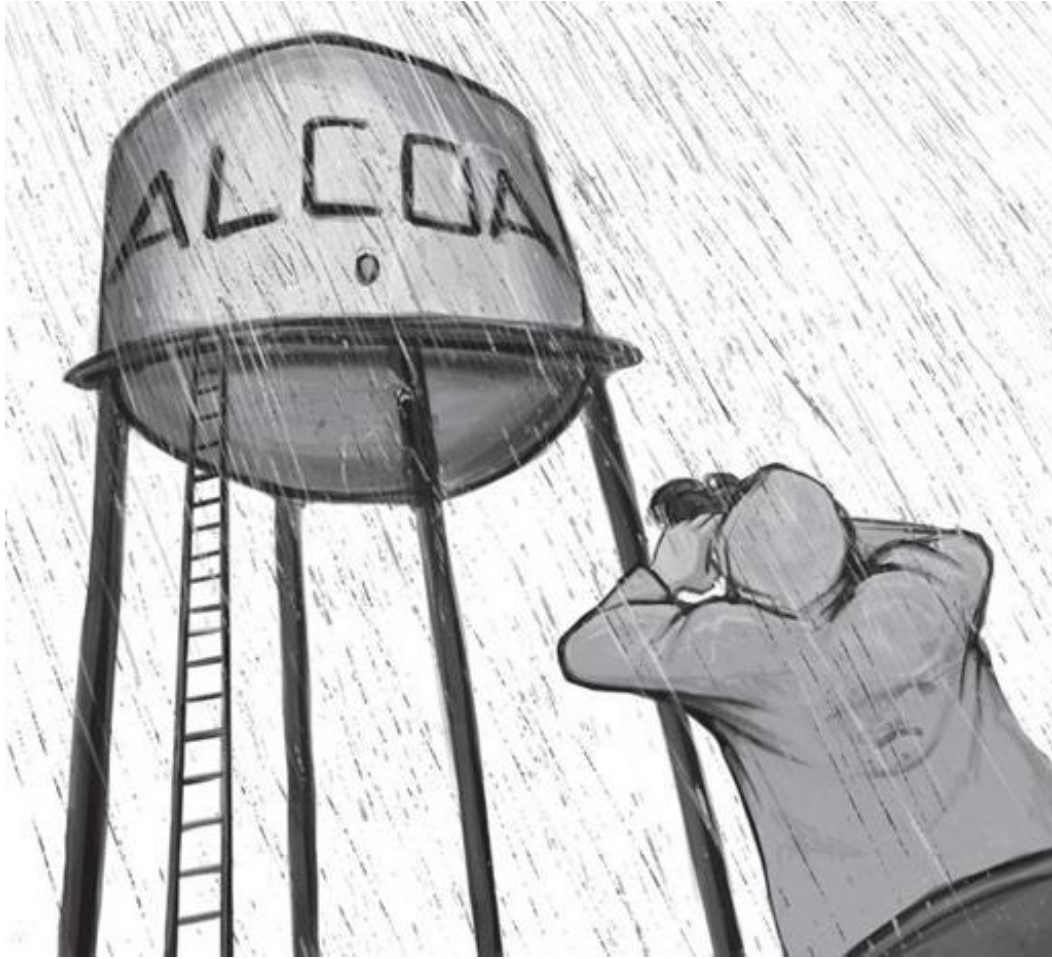
Everything began to change after that. The unions, which normally fought with Alcoa over one-year contracts, signed an unheard-of seven-year deal. Reporting injuries became mandatory, and a real-time safety information system was built to support this function. Management was given twenty-four hours to report any injury and another twenty-four hours to present a plan for how that injury would never occur again at *any* Alcoa plant. One senior manager in Mexico was fired after failing to report an event in which two workers went home ill. It turned out that what had made those workers ill on the job made many more workers ill the next day. But rather than reporting the initial incident, it was covered up and that was unacceptable in the new Alcoa.

Workers became more agile as they made their workplace safer. One dangerous job involved climbing a ladder to record data from gauges positioned near the top of water tanks. The data was collected every day, even in rain, snow, or sleet.

After being empowered to make safety a prerequisite, workers came up with a solution: buying high-powered binoculars so they could record the data from the ground. This also made the work quicker, easier, and far more graceful.



A *Wall Street Journal* reporter once asked Mr. O'Neill if all of this talk about safety at Alcoa was really true. O'Neill told the journalist not to trust him but to go see for himself. The journalist drove to Alcoa's Davenport, Iowa, plant, on the shores of the Mississippi River.



He parked his car in a parking lot. It was pouring rain, and since he didn't have an umbrella, he began sprinting toward the nearest building entrance. In the middle of the parking lot, he was startled when he heard someone yell, "Stop!"

The journalist immediately stopped as an hourly worker ran over with an umbrella and said, "I don't know who you are, but we really care about safety here. It's wet, the pavement's slick, you're likely to fall and hurt yourself, and we don't let that happen here at Davenport. So let me take you into the building."



The journalist later called O'Neill and said, "I think your safety thing is working."²

Safety transformed Alcoa. The company's stock price soared, injuries and lost workdays steadily declined, quality improved, and innovation flourished. O'Neill had succeeded beyond everyone's wildest dreams in returning Alcoa to its former glory. Harvard Business School wrote case studies about the transformation and called O'Neill a business maverick.³

In 2000, when O'Neill left Alcoa after thirteen years as CEO, safety continued to improve, the company kept innovating, and the stock price continued to climb. O'Neill had fundamentally changed the DNA of a one-hundred-year-old industrial giant based on an unwavering commitment to worker safety.

Safety is a doorway to excellence. Make safety a prerequisite.

¹. Paul O'Neill Sr., Ken Segel, and Geoff Webster, *A Playbook for Habitual Excellence: A Leader's Roadmap from the Life and Work of Paul H. O'Neill, Sr.* (Pittsburgh: ValueCapture LLC, 2020), 13–14.

². hsi.com/blog/3-questions-after-the-greatest-safety-speech-ever-given.

³. store.hbr.org/product/workplace-safety-at-alcoa-a/692042, hbsp.harvard.edu/product/600068-PDF-ENG, hbsp.harvard.edu/product/KS1035-PDF-ENG.



Safety Is Golden

Imagine you are a construction worker, walking the steel beams and wooden planks of a bridge under construction—hundreds of feet up in the sky—with no net to catch you if you fall. To preserve your self-respect, and your job, you hide your fear from coworkers. And for good reason: twenty-four of your coworkers have already died while building this bridge.

That's precisely how one engineer described his job during the 1933–36 construction of the San Francisco–Oakland Bay Bridge. Today, high-steel construction workers have no such fear since safety is an integral part of the work. But back then, accidents happened regularly, and the grim expectation was that one life would be lost for every million dollars spent.

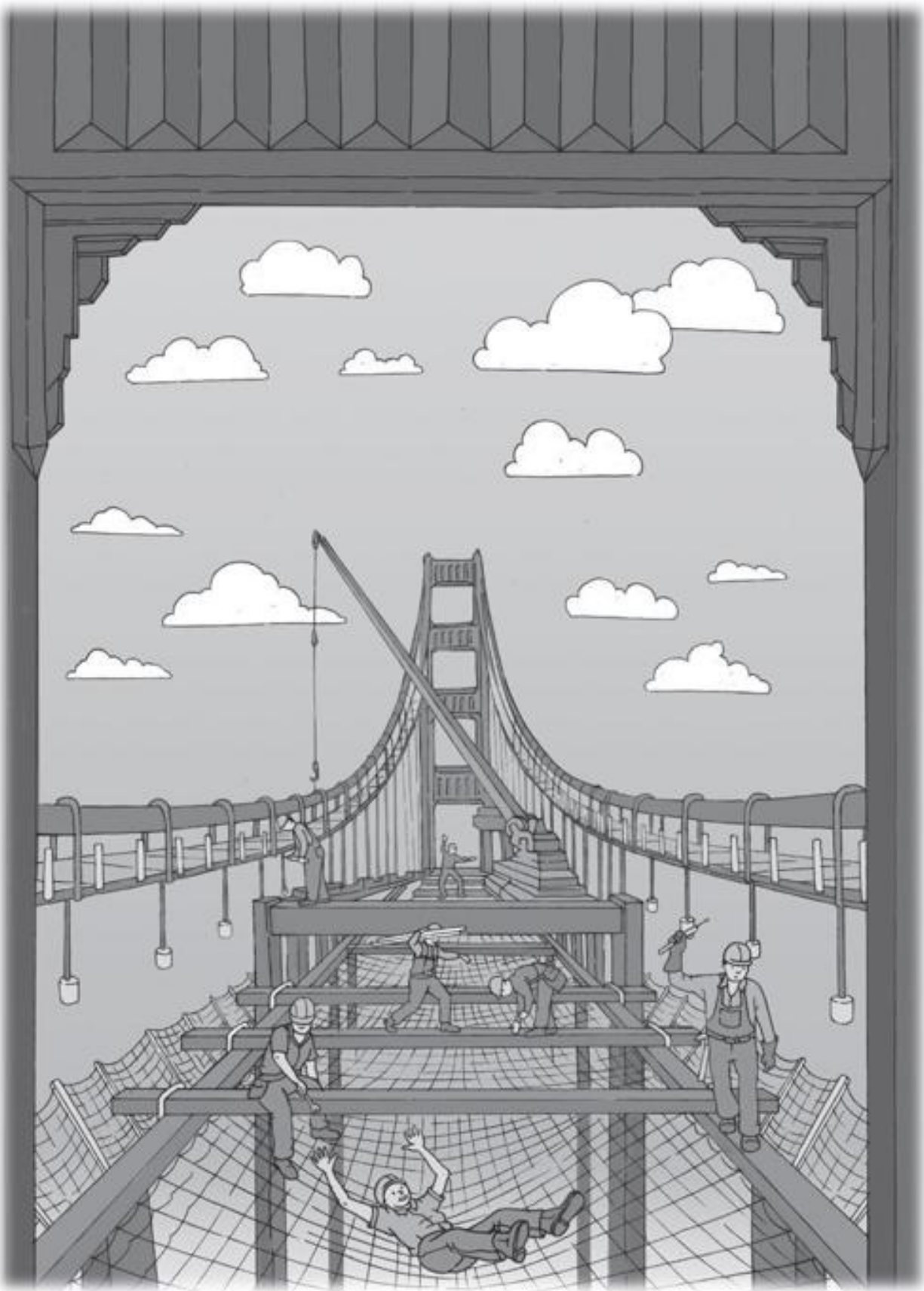
And that's why the Golden Gate Bridge is so remarkable. Built around the same time as the San Francisco–Oakland Bay Bridge, it operated like a modern construction site in its devotion to safety.

Joseph Strauss, chief architect for the bridge, said, “We had the idea we could cheat death by providing every known safety device for workers.” During forty-four of the fifty-one months of construction, there were zero deaths and few injuries.

Strauss and the bridge's resident engineer, Russell Cone, used the latest safety equipment (like glare-free goggles and mining helmets adapted for bridge construction), coinvented new safety equipment (including a respirator to protect workers while cleaning steel), and made safety procedures mandatory. One gruff steelworker was fired immediately when he refused to attach his safety belt to a tie-off line.

Before construction of the bridge's roadway, Strauss envisioned the most “expensive, elaborate safety device ever conceived for a major construction site.” This device was a safety net that would be hung underneath the emerging roadway. At \$130,000, it was a precious sum in the midst of the Great Depression. Strauss lobbied hard, however, and won its approval.

The safety net grew as the roadway grew, extending ten feet beyond the width of the roadway to protect workers at all times. Its presence boosted morale on the construction site. As one worker said, “No matter how high up you were and how hard you'd been blown off, you would still fall into the net.”



Nineteen men who accidentally tumbled into the net but survived their fall dubbed themselves the Half Way to Hell Club.

One historian said, “The loss of life, the delays that would occur from men working slower because they had to be a bit more careful so they wouldn’t fall, probably made the \$130,000 a very economic innovation.”

The excellent safety record was shattered, however, during the final seven months of construction. One worker was crushed by a support beam in October 1936. And on February 16, 1937, a five-ton platform collapsed, ripping through the net and sending twelve men plummeting 220 feet into the frigid waters below. Ten of those twelve workers died from the fall, bringing the total fatality count to eleven.

Without the safety net, that fatality count would have been thirty, since all nineteen members of the Half Way to Hell Club would have also perished. Given its \$35 million budget, thirty deaths during construction would have been close to the 1930s average of one death for every million dollars spent.

Yet injuries and deaths were kept to a minimum during construction of the Golden Gate Bridge because safety equipment, training, procedures, and oversight were an integral part of the work. It resembled a modern construction site at a time when insufficient attention was paid to safety. Today, the bridge is both an iconic symbol of the San Francisco Bay Area and a landmark for worker safety.

Safety enables confidence, and confidence enables speed. Invest
in protecting people.



Robert Plant and My Ticket Ordering Ordeal

Have you ever had trouble ordering something online? (That was a loaded question, since we have *all* had trouble ordering something online at one time or another!) One evening I was using a popular ticket-sales website to purchase two seats for a Robert Plant concert in Austin, Texas. I was using Google's Chrome browser on a modern iMac to create a new account—a required step in the ordering process. I carefully entered the payment information, but the site said I needed a “Payment Method” and stopped me dead in my tracks.

Huh? I already did that!

I tried two more times, same error. So I waited ten minutes for a Live Chat support person, who told me, “Try IE [Microsoft Internet Explorer].”

“I’m on a Mac!” I replied as it began to dawn on me how incredibly bad the ticket site was. “We don’t have IE here!”

“Then try Firefox,” suggested the Live Chat person.

No cigar—that didn’t work either.

So I decided to log out of my newly created account, log back in, and enter my payment data one more time. I pressed “next,” and I waited—and waited. Five minutes passed while *nothing* happened! I didn’t want to hit “submit” again, for fear of being double-billed and having to deal with *that* mess.

Now I was downright mad—I was done messing around with the website. I found the company’s sales phone number and called to buy the tickets by phone.

By that time, I wasn’t surprised in the least that the support person had to start the order from scratch. He said they had been experiencing a high volume of calls that were making his life difficult. Problems with online ordering don’t just frustrate users, they frustrate support staff as well. During the course of the phone call, I learned that I had nearly bought standing-room-only tickets while using the website, which had not been apparent to me while I was trying to place my online order.

After completing my phone order for two actual seats, the support person asked me if there was anything else he could help me with. I said he could help me by improving their website.

Later, I wondered what this ticket company’s plans might be for future versions of their website. If they planned new features on top of a site that wasn’t functional, something was very wrong.

I wondered what this company’s culture was like. I doubt that safety was a

driving value.

I wondered what was tracked to measure the company's health. If they mostly focused on ticket sales metrics, they were blind to bigger problems.

I wondered if, like many other companies at the time, they were implementing an agile process superficially but not actually inspecting and adapting to make their website safe for users and support staff.

I wondered whether they had invested in training technical staff, or in testing tools, or whether they used any automated browser-testing solutions. They clearly didn't value safety enough to use such tools, or if they did, perhaps they had learned to live with broken tests and random failures.

I wondered about their competitors. Given my experience, I had promised myself I would seek out competing sites. In fact, because the site couldn't even perform the most basic functionality (i.e., allowing people to buy tickets), I was convinced that this company had lost its way.

This company and website:

- wasted an hour of my valuable time,
- irritated me with nonsense error messages and nonfunctioning pages,
- scared me with the possibility of getting double-billed,
- misinformed me by not making it clear that I was in the process of buying standing-room-only "seats,"
- made ordering tickets such an ordeal that I would hesitate to ever buy tickets from them again, and
- angered me enough that I have told numerous friends (and you the reader) about this pathetic experience.

The company made ticket buying unsafe, even for someone with a modern Mac and common browsers. All of this is bad business, indicating a lack of sound leadership and a failure to appreciate the safety of users, support staff, and others.

Step number one on the road to making safety a prerequisite is being aware that you are injuring yourself and others. This leads you to visualize the suffering people are experiencing in order to reduce or eliminate it.

For example, this company ought to keep metrics on how many users have ordering ordeals. They ought to make it crystal clear when you're buying a seat or a standing-room-only ticket; otherwise, their service is unsafe. They ought to follow up with buyers, especially first-time buyers, to see if they had a successful purchasing experience. They ought to follow up after the concert to make sure people were happy with their tickets. Such data ought to be highly visible to everyone within the company, hopefully on a dashboard that everyone

sees several times a day. Most important of all, the company needs to make buying tickets a safe, enjoyable experience.

Learn about and eliminate anything that makes customers unsafe.



Pixar's Near Miss

Animated films, such as Pixar's award-winning hits—including *Finding Nemo*, *The Incredibles*, *Toy Story*, and many more—are composed of thousands of digital files. Characters, objects, backgrounds—everything you see and hear in a completed movie—are stored in separate digital files. It takes dozens of people working several years to carefully integrate the thousands of files into a final, feature film.

So imagine what would happen if those files were suddenly erased.

That disaster actually occurred during the production of *Toy Story 2*. As Pixar cofounder Ed Catmull describes in *Creativity, Inc.*, a well-meaning employee in the systems group was innocently deleting the contents of a hard drive, without knowing that it stored *all* of the files for *Toy Story 2*! Ninety percent of the movie had been deleted before a technical director realized what was happening and got the systems employee to halt the process.

Fortunately, Pixar made data backups, so they could simply restore the deleted files, right? Wrong! Unfortunately, their backup process wasn't working.

Now they were truly ready to panic! The lost digital artwork would take thirty employees a year to re-create. And since Pixar had recently become a publicly traded company, the loss and the delay it would cause could be catastrophic for the company's stock value.

By some twist of fate, the movie's supervising technical director, a woman named Galyn Susman, happened to have a backup of the movie on her home computer. She'd recently given birth, and without getting formal permission, she had set up a weekly backup process to her home computer so she could work remotely. She and a colleague drove to her house, wrapped the computer in blankets, and very delicately brought it back to Pixar's offices, where the files were restored to a hard drive. *Toy Story 2* was saved, and a near disaster was averted.

So who was accountable for this dangerous near miss? Ed Catmull knew that blaming or penalizing people would not help anyone, especially in a company that empowered people to solve their own problems. He said,

Our people have good intentions. To think you can control or prevent random problems by making an example of someone is naïve and wrongheaded. Moreover, if you say it is important to let the people you work with solve their own problems, then you must behave like you mean it. Drill down, certainly, to make sure everyone understands how important it is that we strive to avoid such problems in the future. But always—always—

*walk your talk.*¹

Pixar had clearly missed opportunities to ensure the safety of its valuable digital assets. Precautions were needed to make sure backups worked, hard drives weren't accidentally deleted, and other unforeseen risks were addressed. The near loss of *Toy Story 2* became an opportunity for staff to increase the safety of their processes and procedures rather than a blame game or an excuse to decrease employee autonomy.

Increase safety rather than blaming or penalizing people.

¹. Catmull and Wallace, *Creativity, Inc.*, 201.



The Shocking Truth About Those Brown M&M's

Big hair, loud music, and elaborate theatrical performances emerged on the hard rock music scene in the '70s and '80s. Mega-popular bands paired their big musical hits with large, complex lighting and stage designs that even veteran roadies found to be a challenge to set up, break down, and move to the next concert venue night after night.

These complex rigs were ripe for disaster. Band members, crew, and sometimes even fans would get hurt or killed as stages collapsed, giant video screens swung loose, and pyrotechnics exploded unexpectedly or ignited deadly fires.

Consider the example of KISS lead guitarist Ace Frehley. In front of a packed auditorium at the Lakeland Civic Center in Florida in 1976, Ace grabbed a metal railing along a staircase that led down to the main stage, making a perfect electrical circuit between the uninsulated railing and his electric guitar and giving him a severe shock. Unable to move due to the current flowing through his body, Ace finally managed to break free—falling several feet onto the hard floor below. His miraculous survival prompted the band to immediately invest in wireless guitar systems to ensure that their instruments were shock-proof.

Aside from occasional accidents such as Ace Frehley's, safety was usually an afterthought for most bands. However, Van Halen and its lead singer, David Lee Roth, were different. Roth possessed a keen sense of how making safety a prerequisite would ensure that his band could perform outrageous concerts that would blow fans away—all without ever putting the band, crew, managers, or fans in harm's way.

The elements that made Van Halen concerts spectacular also made them extremely dangerous. The band used more PAR (Parabolic Aluminized Reflector) lights than any band in history. Safely setting up 850 huge, theatrical PAR lights—along with the rest of the stage and the band's instruments, amplifiers, and sound equipment—was a complex affair. And doing it in auditoriums and arenas built in the '50s and '60s—with their substandard electrical wiring and flooring unsuited for thousands of pounds of equipment—was even more challenging.

So how did Roth manage to ensure disaster-free performances?

First, while most bands at the time had a brief contract rider with

performance terms and stage requirements, Van Halen had a phone book–sized rider. It contained detailed specifications about the lighting and stage design, including voltage requirements, spacing between electric sockets, load-bearing details such as the exact weight of the enormous stage, and so on. If a promoter failed to follow these instructions, the band, crew, managers, and even the fans could be injured or killed.

Second, the forward-thinking and clever Roth had the band’s lawyers insert a tiny clause into the voluminous contract rider, specifically designed to test whether promoters actually read the detailed specifications. As Roth tells the story, “It was part of the Van Halen contract as we toured through the arenas in the ’80s that there would be no brown M&M’s found in the backstage area, or the promoter would forfeit the entire show at full pay.”¹ Rumors about Van Halen’s ban on brown M&M’s began to circulate, decrying the band’s privilege and excess. Little did they know about the band’s commitment to safety.



When the band showed up to a venue, Roth immediately checked the catering tables, looking for the required bowls of M&M’s. If he found brown ones in the bowls, he knew that the venue was potentially unsafe because the

promoter had not carefully read the contract. Serious line checks were then conducted by the band's crew to find and fix any safety concerns—before and not during the performance. This brilliant plan made safety a prerequisite at Van Halen concerts.

Don't leave safety to chance. What are your brown M&M's?

1. [youtu.be/ IxqdAgNJck](https://youtu.be/IxqdAgNJck).



Amy Edmondson's Eureka Moment

When Harvard Business School professor Dr. Amy Edmondson was a first-year doctoral student, she had the great fortune to join an interdisciplinary team of researchers studying medication errors in hospitals. It was the perfect place for Amy since she was interested in studying teamwork and its relationship to mistakes. The research team, which included experienced physicians and trained nurse investigators, studied eight medical units in two hospitals every couple of days for six months.

Amy hypothesized that the most effective teams would make the fewest errors. A typical error would be an incorrect medicine dosage for a patient. The research team would investigate how many errors were made by a patient care unit, composed of doctors, nurses, and clerks. Since Amy wasn't a medical expert, she relied on the research team's physicians to judge whether human error had occurred and nurse investigators to review charts and interview caregivers to obtain error rates. Amy added to this data with a team diagnostic survey—a validated instrument administered to everyone within each unit to determine which teams were performing better than others.

After six months, Amy studied the error rates and survey data to see if they matched her hypothesis. She was thrilled to find a “10-fold difference in the number of human errors per thousand patient days (a standard measure) from the best to the worst unit.”¹ But something was wrong. While the data showed large differences between teams, it turned out that the most effective teams had the highest error rates!

What?!? Could there have been a mistake? How could teams with better collaboration and teamwork make more mistakes?

In *The Fearless Organization*, Amy describes how she tried to make sense of this data:

I thought about the need for communication between doctors and nurses to produce safe, error-free care. The need to ask for help, to double-check each other's work to make sure, in this complex and customized work environment, that patients received the best care. I knew that great care meant that clinicians had to team up effectively. It just didn't make sense that good teamwork led to more errors. I wondered for a moment whether better teams got overconfident over time and then became sloppy. That might explain my perplexing results. But why else might better teams have

*higher error rates?*²

That question led Amy to her eureka moment and a new hypothesis: “What if the better teams had a climate of openness that made it easier to report and discuss errors? The good teams, I suddenly thought, don’t *make* more mistakes, they *report* more.”³

Amy’s new hypothesis would ultimately help her discover the critical importance of psychological safety in teams. She hired a new research assistant who was “blind” to previously collected data and unaware of Amy’s new hypothesis. Amy asked the researcher to quietly observe teams, conduct open-ended interviews about the work environment, and collect data on whether people felt able to talk about mistakes and report errors.

The data he collected matched perfectly with the previously collected error rates. Amy wrote, “People in the better teams (as measured by my survey, but unbeknownst to the research assistant) talked openly about the risks of errors, often trying to find new ways to catch and prevent them.”⁴

Amy’s work in the hospital helped her understand the importance of the interpersonal climate in teams. Within a few years, she would call this difference “psychological safety” and continue to research how it impacted teams in many industries.

High-performing teams operate within a psychologically safe climate that allows people to be comfortable discussing and reporting errors, explore ways to remove errors, and continuously improve.

¹. Edmondson, *The Fearless Organization*, 2018: 9, 10.

². Ibid, 10.

³. Ibid, 10.

⁴. Ibid, 11.



Is It Safe to Fail?

Alan Chedalawada, an experienced management consultant and expert in lean methods, wondered whether employees were afraid to communicate negative news to management within a department of a Wall Street bank. He observed that status updates for most of the initiatives always had a positive spin. Executives dismissed his concern, explaining that people in the firm could communicate openly and were *safe to fail*. If status updates were positive, they explained, then it was because the initiatives were going well.

The executives had a dashboard for all current initiatives that considered various factors, such as whether the initiative was within budget or whether it was going to meet its expected release date. Based on these criteria, each initiative got a color:

- Green meant the initiative was in good shape.
- Yellow meant the initiative was experiencing some challenges, but was otherwise okay.
- Red meant the initiative was struggling.

The three-color status scheme was eventually changed to allow for even more positivity. With the change, an initiative could be marked yellow-trending-green or red-trending-yellow.

One day, Alan encountered a key initiative that was clearly experiencing challenges and should be considered yellow. He wondered how management would react if the initiative went red. After Alan made a few changes to some data, the initiative showed up red on the executive dashboard.

It didn't take long for the executives to freak out. However, instead of asking how they could help or what could be done to get the initiative back on track, they yelled at people, demanding to know: "Who is responsible and how did this happen!?"

After observing their behavior, Alan met with the executives and revealed his ruse. They were not happy with his little stunt, but they acknowledged that he was correct in proving that they did not handle failure well. The executives were clearly not fostering a psychologically safe environment in which people felt safe to reveal the true status of their work. It was an important lesson for them to learn.

Harvard's Amy Edmondson, author of *The Fearless Organization*, makes it abundantly clear how psychological safety is essential to high performance. When there is a culture of fear, people are reluctant to admit mistakes because they are afraid of retribution. And that's when people either remain silent or spin

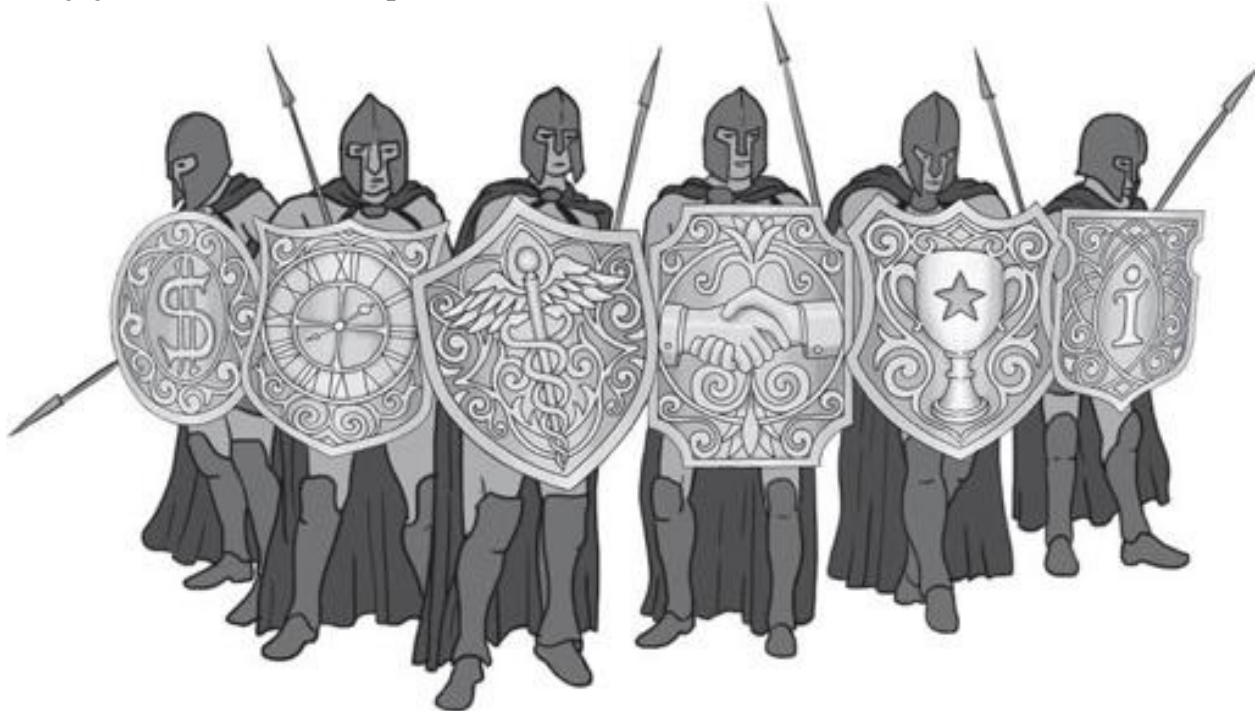
the truth. Alan sensed this and helped the bank's leaders improve the psychological climate.

Is it safe to speak the truth in your team, your organization? Do you dwell in a culture of fear? If problems or failures are most often met with punishment or blame, do people understand how that negatively impacts productivity and innovation?

How safe is it to fail in your world? How could you test it?

● ● ● Never Lose Your Shield!

Ancient Greek soldiers who fought at the front of the battle lines were called *hoplites*. The most important piece of equipment they carried with them into battle was their shield. Spartan warriors were taught that they could lose their helmet, breastplate, or other armor, but they would be dishonored if they lost their shield because “You wear the armor for your own protection, but you carry your shield for the protection of the entire line.”



What is the equivalent of a shield in your work? I once worked at a New York bank that suffered a major blow when a trader got caught engaged in illegal activity. His work so tarnished the reputation of the bank that the stock price tumbled and they were soon acquired by a competitor.

Like a shield protecting the entire hoplite line, an organization's integrity and reputation is something everyone must protect. When only one individual in an organization is the sole expert in a critical area of a product or service, the organization is at risk. Should that individual quit or become too ill to work, no one will be able to easily and quickly take over their work. Good management guards against this outcome by actively encouraging staff to avoid letting knowledge silos develop.

In your own work with colleagues, be keenly aware of what you protect for yourself and what you protect for the larger group.

Never lose your shield.



A Virgin Derailment

One evening in February 2007, a train with 105 passengers traveling from London to Glasgow derailed.

The train, owned and operated by Sir Richard Branson's company, Virgin Trains, was traveling at the authorized speed of ninety-five miles per hour when a faulty track switch caused it to jackknife and plunge down an embankment.

While many fatalities were expected in the aftermath of this devastating accident, only one person died, an eighty-four-year-old woman who passed away from heart failure hours later in the hospital. Twenty-eight passengers and two (of the four) crew members were seriously injured. When Richard Branson arrived at the crash site, only eleven remained in the hospital, none of whom had life-threatening injuries. The most seriously injured person was the train's driver, who suffered a broken neck. A doctor who evaluated him said he would recover.



Some called the crash a miracle because so few people died or were seriously injured. Yet, it was no miracle.

When Virgin Trains was first conceived, Branson and his team insisted on investing in every possible safety innovation. They wanted every passenger on a Virgin train to be able to walk away from a crash—even if the train was traveling at full speed. Virgin trains were designed with numerous safety innovations, including unusually strong one-piece aluminum structures built into the carriages to prevent loss of survival space in the event of a crash. Interior lighting was designed to stay on after an accident—using specifically designated batteries—

so people inside would be able to see after a crash. Couplers (linkages between carriages) were designed so that carriages would not ride up on top of each other in a crash. Laminated carriage windows—like those used in automobiles—were installed to prevent flying shards of glass from killing or injuring people.

Branson said, “The train is built like a tank. They perform beautifully and it has a system where we can check every part of it by computer.”

A UK Rail Accident Investigation Board report on the accident identified twenty-nine safety recommendations, almost all of which were directed at the company that maintained the train tracks. Referring to the Virgin train, the report said,

While objective comparisons cannot be made because of the different trains, speeds and forces involved, overall, the crashworthiness performance of the [Virgin Train] avoided, almost completely, a number of hazards. These include multiple ejections through windows, loss of survival space and penetration of the passenger compartment by external structures, all of which have been known to cause fatal and serious injuries in the recent past.¹

In fact, no windows were broken, and the carriages did not crumple as a result of the impact. Branson said, “I think if it had been any other type of train, the injuries and mortalities would have been horrendous.”

Safety is not a miracle; it’s a choice.

¹. UK Rail Accident Investigation Board, “Rail Accident Report: Derailment at Grayrigg,” February 23, 2007, assets.publishing.service.gov.uk/media/547c9037ed915d4c0d000199/R202008_081023_Grayrigg_v5.pdf.



Heroism Is Hazardous

When Steve Kuo, a lean/agile coach, first joined a large wireless phone company, he asked if anyone wanted to program with him so he could learn his way around the code. A senior programmer volunteered. A few minutes later, they sat down together to fix a defect. The programmer opened a file and began scrolling through a two-thousand-line function. The programmer quickly spotted the issue and changed an “AND” to an “OR” on one line of code.

Steve asked whether there were any automated tests to confirm that the change was safe. The programmer replied, “We do not have any automated tests—that’s what our testers are for. Besides, this is a simple change and I know that’s the issue.” The programmer checked in the new code, the testers found nothing wrong with it, and the fix was pushed to the production computers.

All’s well that ends well, right? Not exactly.

A day later, the entire department got an all-hands-on-deck emergency call. Salespeople in the wireless stores were unable to sell mobile phones due to a new defect that had just emerged, and the company was losing hundreds of thousands of dollars every hour due to this problem. The senior programmer, with whom Steve had worked the previous day, volunteered to help.

A small crowd of programmers gathered around a computer. The senior programmer opened the file, scrolled to the line of code he had changed the day before, and said, “I see the problem!” He quickly reversed the change he’d made the day before, turning the “OR” back to an “AND.” The “fix” was immediately put into production, and it solved the problem. Everyone cheered him for his brilliant work.

Steve was horrified.

The “hero” of the day had actually caused the problem—losing the company hundreds of thousands of dollars in revenue—due to overconfidence, hurrying, insufficient testing, and no real safety. It was a dangerous way to work, like a high-rise construction worker hanging down from a beam without any protection.



Steve didn't get angry at the programmer or call him out publicly for the unsafe way he approached the problem. Instead, he privately asked the programmer what governed his thinking when he changed the code. The programmer explained that he felt safe to make such changes because of his seniority within the team and the fact that his teammates looked up to him. That response helped Steve understand that the programmer had a lot more to learn about safe and effective software development.

Over the next few years, Steve taught the testers and programmers how to

program in pairs and ensembles rather than alone, how to design software guided by automated tests, how to continuously integrate and deploy code, and how to visualize the work to increase flow and reduce bottlenecks. The lean and agile practices that Steve introduced enabled this software department to work quickly while under control. They had once been the butt of all the quality jokes, but they eventually became the envy of the entire organization.

Replace heroics with agility. Heroism is hazardous.



If It Ain't Broke, De-Risk It

Horried” is perhaps the best word to describe what Robert Benefield, a software consultant, felt when in 2011 he first glimpsed the single ancient computer that was being relied on to calculate how much electricity power plants spread across several European countries should generate. It was a rare model of the PDP-11 minicomputer—a model that hadn’t been manufactured since the early 1970s. DEC (the Digital Equipment Company), which built it, had long been out of business. Robert realized that if any part of this ancient computer were to break, he’d likely need to raid a museum exhibit at the Computer History Museum in Mountain View, California, to hunt for parts.

The software running on the archaic machine was written in an even older language called FORTRAN. It produced hourly calculations, twenty-four hours a day, seven days a week, and these calculations were used by power plants, grid operators, and power regulators to manage the production and distribution of electricity. This hourly data was produced by analyzing large amounts of information about weather, electricity usage patterns, fuel costs, and so on. Robert discovered that this software had no documentation and that the last programmer who understood the code was about to retire!

To make matters worse, the amount of data the software had to traverse was becoming so large and cumbersome that the system sometimes failed to generate hourly data on time. When a run took more than sixty minutes to complete, the power stations would have to guess how much electricity to generate for that hour. This could lead to serious problems such as brownouts and spikes, which could potentially take down the entire electrical grid. Regulators fined the company when this happened, and as these fines multiplied, the operator’s license was threatened.

Despite all of these risks, management had no plans to make any changes to the system. Their philosophy seemed to be: If it’s not broken, why fix it?

Surprisingly, Robert had not been brought in by the company to help with these issues. Yet, since his project did involve getting data both into and out of the PDP-11, he could not help but explore the many risks associated with it. And the more he looked, the more he knew that this extremely fragile, outdated solution needed to be modernized.

So, taking matters into his own hands, he carefully ported the old FORTRAN code to run on a new Linux personal computer (PC). When he was done, the program performed the same job in under a second! Modern computers are many orders of magnitude faster than 1970s machines.

Robert showed the solution to the company's leadership team—they couldn't believe it ran in under a second! The FORTRAN code running on the PDP-11 took more than fifty minutes to do the same task, sometimes longer. How could this new solution, running on a mere PC, run so much faster and still be correct? The facility managers were skeptical and refused to use it.

Robert advised them to simply run the new software every hour simultaneously with the existing solution and compare results. If the results were consistently identical, then they would feel safe to switch to the modern solution. Management agreed to give it a try.

Sure enough, after three months of hourly checks to see that the old and new solutions produced identical results, the facility managers were finally convinced, and they decided to make the switch. Robert breathed a sigh of relief. While he was now the only person familiar with the code, at least the system no longer ran on severely outdated hardware and software. A potential continent-wide disaster had been averted.

Just because something works doesn't mean it will continue to work. Look for risks and remove them safely.



Pre-Work Hazard Analysis

A few years ago, I participated in a class on safety culture and was the only software person among nuclear, electric, automotive, and manufacturing engineers. They all kept using the term *tailboarding* and I finally had to ask what it was. An engineer from an electricity company in Austin, Texas, explained: a “tailboard” is a short, informal dialogue conducted by workers just before doing work that poses actual or potential safety hazards.

Safety hazards? We have plenty of those in software development! I immediately knew that tailboarding could help us, so I set out to learn more.

Tailboarding is sometimes referred to as a job briefing, toolbox talk, or a tailgate meeting. Tailboards help people identify and address safety hazards of a specific job before commencing work.

During a tailboard, a group of workers typically spend fifteen to thirty minutes surfacing hazards and deciding how best to eliminate them or address them via safeguards. Better tools can eliminate some hazards, while better procedures can help reduce other hazards.

Tailboarding complements Job Hazard Analysis (JHA). While tailboards are informal, a JHA is a formal written analysis of hazards and safeguards for the job.

If unanticipated hazards are discovered, work must stop, a new hazard assessment must be conducted, and a new tailboard held before work resumes.

After the class, I explained tailboarding to my colleagues at Industrial Logic. In our context, it could help us discover what could interfere with our ability to safely deploy new code to production, what safeguards we could use to better protect ourselves, and what work is simply too risky to perform without sufficient pre-work. We decided to give it a try.

Before releasing a new navigation feature for our eLearning, tailboarding helped us identify hazards in the usability, programming, and release of the new feature. One team member said:

The tailboard helped us make sure that the transition from the old navigation to the new navigation was smooth. We identified over a dozen items that could have easily broken the new navigation. The tailboard helped us work through the list of hazards to make sure we had addressed all major problems before releasing.

Tailboarding is one of many safety practices that help drive out fear. It is proactive, rather than reactive, pre-work hazard analysis that is applicable to any error-prone, stressful, complex, or dangerous work. It can make work more

productive and harmonious.

Proactively analyze hazards and handle them before they harm
you.



Healthy Fear

On August 9, 1896, Otto Lilienthal, a German engineer and aviation pioneer, was flying his latest glider at an altitude of fifty feet (fifteen meters) when a strong gust of wind suddenly threw his craft into a nosedive. Unable to regain control, Lilienthal crashed to the ground and broke his neck. The following day he died in a Berlin hospital. He was just forty-eight years old.

News of Lilienthal's death shocked people around the world, especially aviation enthusiasts. He was the first person in history to make repeated, successful glides—some of which covered more than one thousand feet and lasted as long as fifteen seconds. Photos of him soaring through the air graced newspapers and magazines around the globe, earning him the nickname the “flying man” and making him world famous.

Unlike the daredevils, cranks, and lunatics who had no business ever attempting to fly during the nineteenth century, Lilienthal was well suited to the task. As a young man, he studied engineering, became a professional mechanical engineer, and later an inventor and entrepreneur. His company, which made small steam engines, gave him the financial freedom to pursue his passion for aviation.

During the 1880s, Lilienthal conducted a scientific study of how birds fly, explored how wings generate lift, tested surfaces to determine the best shape for wings, compiled tables of aerodynamic coefficients, and performed numerous unmanned glider experiments. He published his findings in an 1889 book called *Birdflight as the Basis of Aviation: A Contribution Towards a System of Aviation*.

By the early 1890s, Lilienthal was ready to test his aerodynamic principles via manned flights. From 1891 to 1896, he completed more than two thousand flights in sixteen glider designs. In 1894, Lilienthal built *Fliegeberg* (Fly Mountain), a forty-nine-foot-high, conical earthen hill that allowed him to launch his glider into the air from whichever direction the wind was blowing. To become airborne, Lilienthal hoisted his glider on his shoulders and either ran down the hill or jumped from a platform. Once in the air, he shifted his lower body and legs to navigate.

Unfortunately, Lilienthal's method for navigating his glider was fatally flawed. When a gust of wind sent his glider into a nosedive, simply shifting his lower body and legs did not enable him to regain control.

Today, Lilienthal is known as the “father of gliding experiments” and one of the most important pioneers of aviation. His tragic death in 1896 reignited Wilbur and Orville Wright's childhood interest in flying and inspired them to

work on solving the problem of flight. The brothers studied Lilienthal's writing, used his data in their early glider designs, and followed his lead in conducting outdoor experiments. Wilbur Wright said, "If you are looking for perfect safety, you will do well to sit on a fence and watch the birds; but if you really wish to learn, you must mount a machine and become acquainted with its tricks by actual trial."



Yet perhaps the most important thing the Wright brothers learned from Lilienthal was healthy fear. Flying was dangerous; it could be deadly. Even experts had blind spots and could be seriously injured or killed. To succeed, they needed to be meticulous in ensuring their own safety during manned flight. Healthy fear influenced the Wright brothers' glider and airplane designs, their decisions about where to fly, and the manner in which they conducted their many experiments to produce the world's first airplane. Years later, when huge crowds came to see them fly, healthy fear led them to cancel flights due to poor weather.

Healthy fear has its place. Pay attention to it.



Be Careful (But Don't Play It Safe)

For some people, *safety* is a trigger word that means not taking *any* risks or always “playing it safe.” In reality, you’ll never accomplish much if you always play it safe. And that’s not the meaning of Make Safety a Prerequisite. Not taking risks is a recipe for failure and stagnation.

Playing it safe often means being *too* careful. If you’re trying to learn to roller skate, being too careful won’t help.

In *Managing the Risks of Organizational Accidents*, James Reason used a battle between France and England to illustrate how being too careful with our defenses can sometimes endanger us. During the Battle of Agincourt in 1415, French soldiers wore extremely thick plate armor, while English soldiers were lightly clad with protection about one-fifth the size of the armor worn by the French force.



Some treacherous terrain caused two groups of French soldiers to bunch together in a field. Once they were within range, the English archers hit them with a flurry of yard-long, steel-tipped arrows. While many Frenchmen died immediately in that attack, those who fell off their disabled horses lay helpless, unable to get back up due to the heaviness of their own armor! The English then made quick work of them.



James Reason noted that “defenses designed to protect against one kind of hazard can render their user prey to other kinds of danger, usually not foreseen

by those who created them, or even appreciated by those who use them.”

We have to find the right balance between being careful and not playing it too safe. Sometimes that involves taking calculated risks. For example, one of my daughters enjoys climbing trees. She’s careful when she climbs. She hasn’t hurt herself yet. One day, she might hurt herself, yet that doesn’t mean she ought to play it safe and stop climbing trees. Her risk of injury is low. Her skill in climbing, combined with the size of trees she climbs, makes her tree climbing an acceptable risk for her and me.

Erik Hollnagel, a safety expert, author, and pioneer of resilience engineering, once said, “Safety is freedom from unacceptable risk.”

Being careful versus playing it safe is always a choice. How much care is needed in the given context? What are the consequences of failing? What level of protection is appropriate? What risks are worth taking? What would need to change to not play it safe?

Be careful (but don’t play it safe).



Silence versus Voice

Katherine Johnson was a pioneering mathematician and one of the first African American women to work at NASA. She knew how to calculate parameters for space travel better than most of the engineers with whom she worked. From February to May of 1958, those engineers would give a series of seventeen lectures to peers at NASA, which included data charts and mathematical formulae that Katherine helped them prepare. But she wasn't allowed to attend the group meetings.

And that bothered her.

She wanted to understand the answers the engineers were trying to get and she knew that the “the real work of refining, testing, and vetting the engineers’ research took place in their group meetings.”¹ For that reason, she wanted to be “in the room with the engineers, absorbing it all, asking questions in real time, not after the fact, as someone gave me the digest version of what happened.”

So she asked to attend the meetings.

Every guy in the room was silent.

Then one of them said, “Girls don’t go to the meetings.”

Not giving up, she asked if there was a law against it.

There was no law. It had just been a practice for as long as females had helped with computations at NASA and its predecessor organization.

After several more awkward moments of silence, the boss allowed her to attend the meetings.

People who “play it safe” in organizations often remain silent when they see problems or disagree with someone (especially a boss). Why would they do that? In this table from her book *The Fearless Organization*, Amy Edmondson explains one reason why.

	Who Benefits	When Benefit Occurs	Certainty of Benefit
Voice	The organization and/or its customers	After some delay	Low
Silence	Oneself	Immediately	High

In other words, by playing it safe and not speaking up, an individual gets an

immediate and guaranteed benefit, robbing the organization and/or its customers of a potential improvement. This isn't always the fault of the individual. When speaking up is risky to one's job (or bonus or status), silence prevails.



After British Petroleum's *Deepwater Horizon* oil rig exploded (killing eleven people and spewing oil into the sea for years), a United States government Energy and Commerce panel found a major discrepancy between British Petroleum's "stated commitment to safety and what it actually was willing to invest." They learned that workers who "came forward with concerns after safety were sanctioned (even fired in one case), which quickly shut down the flow of safety-related information." To top it off, the panel learned that workers received a 25 percent bonus to increase production numbers, thereby discouraging reporting on health or safety hazards.

Edmondson elaborated on why silence tends to win:

*Another way to think about the voice-silence asymmetry is captured in the phrase, “no one was ever fired for silence.” The instinct to play it safe is powerful. People in organizations don’t spontaneously take interpersonal risks. We don’t want to stumble into a sacred cow. We can be completely confident that we’ll be safe if we are silent, and we lack confidence that our voices will really make a difference—a voice-inhibiting combination.*²

In extensive interviews with employees at one company, Edmondson found that people were “missing out on the satisfaction of the chance to act on their ideas and create change. Instead of helping to create a learning organization, they were just showing up and doing their jobs.”³

Katherine Johnson used her voice to go against a discriminatory group practice, make her work more satisfying, create a path for other women to contribute, and help the United States make progress in the pursuit of space travel.

Think hard before remaining silent. Your voice matters.

¹. Katherine Johnson, *My Remarkable Journey, A Memoir* (New York: HarperCollins, 2021).

². Edmondson, *The Fearless Organization*, 2018.

³. Edmondson, *The Fearless Organization*, 2018.



The Cinder Block Incident

Andy, the CEO of a forty-person startup, was stressed. The company needed to raise its next round of funding and the board of directors would be visiting the office the following week. Andy thought that the office, which was crowded with desks from new hires, looked sloppy. What would the board think? Suddenly, Andy looked out of his second-story window and spotted Helen pulling a cartful of cinder blocks across the parking lot and toward the office. What was she doing?

Helen was an agile coach and longtime colleague of Andy. They'd worked together at a previous startup, and both were among the first ten employees at the new company. Helen had just returned from a local home-improvement store when Andy saw her in the parking lot.

The company had recently hired Samantha, an engineer, and Helen met with Samantha to ask what she needed on her first day of work. Samantha requested a standing desk since her lower back sometimes bothered her. The company already had four standing desks, none of which were the expensive, motorized kind. Instead, they used inexpensive desks, propped up on cinder blocks. Not an elegant solution, but it was in keeping with the startup's lean culture. Helen figured she'd buy cinder blocks for Samantha's new desk and have a few spares left over for future desks.

In the middle of the parking lot, Helen's phone sprang to life. She saw a text message from Andy, in all caps: "WHAT DO YOU THINK YOU'RE DOING? DON'T BRING THOSE CINDER BLOCKS INTO THE OFFICE!!"

Suddenly, Andy appeared in the parking lot and began shouting at Helen at the top of his lungs: "YOU'RE MAKING THIS OFFICE LOOK LIKE A DORM ROOM! WHAT THE FUCK ARE YOU DOING!? TAKE THOSE BACK! I DO NOT WANT TO COME DOWNSTAIRS AND SEE ANY MORE CINDER BLOCKS IN THIS OFFICE!"

Andy's face was beet red. He departed as quickly as he had appeared. Helen didn't move. Her heart was beating fast. What would she do about Samantha's standing desk? She decided the correct solution was to continue with her plan. Helen pulled the cart of cinder blocks inside the office, found Samantha's work area, and helped prop up her desk with the cinder blocks. Samantha was thrilled.



Just then, Andy's text messages lit up Helen's phone again. The last one said, "COME TO MY OFFICE IMMEDIATELY!"

After putting the remaining cinder blocks in storage, Helen walked upstairs toward Andy's office. Again, her heart was pounding. Once inside Andy's office, Helen could see a vein popping out of Andy's face, near his left temple. He was irate. "I want you to return those cinder blocks and bring me a receipt for them by 5 PM tomorrow," he demanded in a stern voice.

Helen was at a loss for words. As she walked out of Andy's office, her mind raced. Why did she bring those cinder blocks into the office? Why didn't she just do what Andy wanted? She could not sleep that night. How could she tell Samantha that she had to take back the cinder blocks? It was Samantha's first

week at the company and that would surely make a bad impression.

The next morning, feeling exhausted and uneasy, Helen told Frank—her boss and CTO—about the cinder block incident. Frank said he'd talk to Andy and try to patch things up. Helen didn't fear for her job, but she was afraid. Frank talked with Andy and then told Helen that the cinder blocks could stay! He'd convinced Andy that they were merely an interim solution.

Two weeks passed. The incident lingered in Helen's mind, and she was mad. She'd known Andy for more than a decade, spanning two separate companies. At the previous company, he'd helped her get promoted. So why did he have to scream and curse at her? And why had he not reached out after the incident? How insulting! There was no closure, and she didn't like leaving things unresolved.

When she was ready, Helen texted Andy, asking him to go for a walk. During the walk, the two patched things up. Helen explained that she was just trying to make the company's new hires happy, and she apologized for the cinder blocks and the makeshift standing desks.

Andy didn't apologize formally, but he did tell Helen that on the day of the incident, he was dealing with a troubling issue in a different part of the company. Helen assumed that his outburst was really about *that* issue, not the cinder blocks. Andy explained that members of the board of directors were going to be doing a walk-through of their office and he didn't want it to look like a college dormitory. He cared a lot about design, he explained, and his own home had even been featured in a local magazine for its architecture and décor.

Joseph Grenny, coauthor of the influential book *Crucial Conversations*, said, "The health of an organization is measured by the lag time between when you feel it and discuss it."

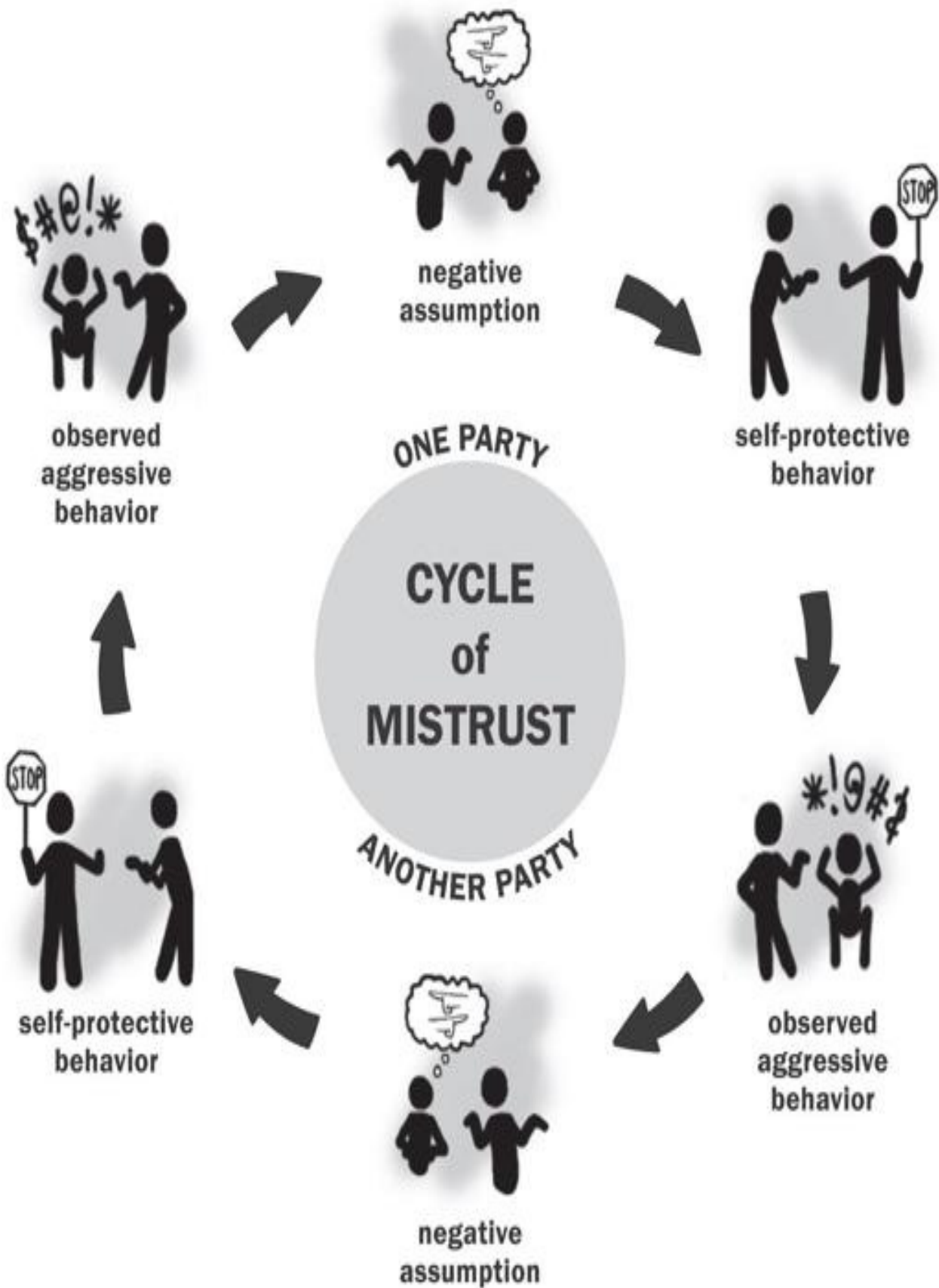
Their walk-and-talk discussion helped Andy and Helen resolve the issue of the cinder blocks. While it wasn't a perfect reconciliation (Helen hadn't felt comfortable enough to tell Andy how awful and disrespected she felt when he yelled at her), it was enough to allow them to work together productively for years afterward. Their startup would grow into a highly successful company.

Resolve conflict quickly.



[The Cycle of Mistrust](#)

Kathleen Ryan and Daniel Oestreich, authors of *Driving Fear Out of the Workplace: Creating the High-Trust, High-Performance Organization*, would describe what Andy and Helen in the [previous chapter](#) experienced as a *cycle of mistrust*. Such a problem may develop between two parties (either people or groups) at any one of the six starting points in the cycle below:



To understand this model, let's explore how Andy, the CEO, and Helen, an agile coach, entered into a cycle of mistrust.

- **Negative assumption:** Andy sees Helen in the parking lot as she's bringing cinder blocks to the office, and forms negative assumptions about Helen's intentions, style, or behavior.
- **Self-protective behavior:** Andy sends angry texts and yells at Helen in order to defend against his perception of Helen's potential to harm or hinder Andy.
- **Observed aggressive, confusing, frustrating, or irritating behavior:** Helen experiences Andy's angry, aggressive behavior and interprets it as intended to harm or block *her*.
- **Negative assumptions:** Based on this interpretation, Helen adopts negative beliefs about Andy's intentions, style, or behavior.
- **Self-protective behavior:** Helen ignores Andy's request to return the cinder blocks to defend against Andy's potential to harm or block her work (establishing a standing desk for Samantha).
- **Observed aggressive, confusing, frustrating, or irritating behavior:** Andy experiences Helen ignoring him and interprets it as her intending to harm or hinder him in some way.

One of the worst things about a cycle of mistrust is that it often doesn't stop after one trip around the cycle. As fear, anger, and a deep mistrust between the parties deepen, second and third trips around the cycle occur. When that happens, communication and collaboration become slow, difficult, and ungraceful—the opposite of agile.

The good news is that cycles of mistrust can be broken. Helen, who had worked together with Andy for years, had the courage to ask Andy to go for a walk, discuss what occurred, and clear the air.

But not everyone has worked together for so long and/or is ready to take such a step. In this case, the parties can learn about the cycle of mistrust model and then privately fill out what they believe is happening during the six parts of the model. Then, if it feels safe, the parties can discuss their perspectives, perhaps with the help of a trusted facilitator.

In my experience, most cycles of mistrust are born out of misunderstandings, unfounded assumptions, or weak relationships between parties. In their book, Kathleen Ryan and Daniel Oestreich explain:

We have found that when people understand the dynamics of the cycle and how it can influence behavior, they become both better able to let go of mistrust and more tolerant of feedback from others. It is very easy for anyone getting feedback in an environment of mistrust to take things

personally and be offended by other people, alleging that they have some type of negative motive. But with a thorough understanding of the cycle, these tense situations can be viewed more objectively, and some of what might have gotten under someone's skin can be seen for what it is: a pattern of negative assumptions to be dealt with constructively. The trap can be avoided by seeing dynamics of the cycle work. When this understanding is coupled with an explicit vision of trust, a leader is in a strong place to begin turning around relationships that have become undermined by fear.^{[1](#)}

Effective collaboration requires trust. Work hard to identify and break cycles of mistrust.

^{[1](#)}. Ryan and Oestreich, *Driving Fear Out of the Workplace*, 2008.



Norm Kerth's Safety Poll

In 2001, Norm Kerth—a pioneering, deeply experienced software veteran and consultant—published a book about a practice he called *retrospectives*. In a nutshell, a retrospective is a review of what worked well, what needs improvement, and what might be done to make improvements. Norm's book, *Project Retrospectives: A Handbook for Team Reviews*, is a gold mine of wisdom on how to help teams learn from retrospectives.

In the book, Norm introduced the idea of taking a safety poll. It's actually the second step in a comprehensive, four-step safety check that Norm conducts before end-of-project retrospectives. The poll helps determine how safe people feel to speak up during the retrospective. Norm recommends that retrospective facilitators acknowledge that with managers present in the room, people may not feel safe to speak up. For that reason, it's helpful to anonymously measure the level of safety that people feel. Here's how Norm suggests doing that:

By means of secret ballots, take a vote on safety, using a rating scale of 1 to 5, in which

5 means "Hey, no problem, I'll say anything."

4 means "I'll say most anything, but a few things might be hard to say." 3 means "I'll share some things, but keep a few things to myself."

2 means "I'm not going to say much. Mostly, I'll let other people bring up issues."

1 means "I'll smile, claim everything is great and agree with whatever the managers say. No way will I let them know what I really think."¹

Once the votes have been collected, Norm suggests tallying them on a flip chart or whiteboard. If the tally of 3s, 2s, or 1s is large enough to indicate that most people don't feel safe to speak up, the facilitator and team may decide to conclude the meeting early, since retrospectives require safety to be effective.

When Liz Keogh, a veteran agile coach, conducts the safety poll, she no longer shares the final tally of numbers with her teams. Liz found that by telling everyone in advance that she would *not* be sharing the final tallies, team members felt more comfortable being honest and voting lower. "I take this as a sign that they feel more comfortable being honest," she said. Liz further increases safety by asking a trusted team member to collect the votes. And sometimes, if she feels too closely coupled to a team and doesn't want her agenda to influence outcomes, she asks another coach to facilitate the safety poll

and retrospective.

However you choose to conduct your own safety poll, it's clear that low votes indicate a lack of psychological safety within the team or larger organization. The possibility of raising real issues and solving them increases when people feel safe to speak openly, in an atmosphere that's free from blame, sarcasm, criticism, or humor at other people's expense.

Use an anonymous safety poll to learn whether it's safe to speak up, regardless of who's in the room.

[¹](#). Norman L. Kerth, *Project Retrospectives: A Handbook for Team Reviews* (New York: Dorset House, 2001).



Resiliency via Diversity

When disasters happen, such as Union Carbide's pesticide plant accident in Bhopal, India (which released poisonous gasses into the air, killing more than 15,000 people and contaminating the land), Boeing's 737 MAX crashes (which killed 346 people), or the partial collapse of Champlain Towers South near Miami (which killed 98 people), everyone wants to know what broke or who is to blame.

In his book *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems*, Sydney Dekker makes it abundantly clear that there is no simple answer for disasters like the ones described above. Whatever goes wrong is related to a web of people, parts, and pressures that make up a complex adaptive system. A failure today may be traced back to decisions and actions that happened slowly over years or decades.

As an example, consider Boeing. For most of its history, the company had a quality-engineering culture and was, according to writer Jerry Useem, "an association of engineers devoted to building amazing flying machines." But when it acquired McDonnell Douglas (a smaller rival) in 1997, McDonnell Douglas executives began slowly changing Boeing's culture to be financially motivated and ruthlessly focused on cutting costs. Fast-forward a few decades, and the 737 MAX disasters happened, killing hundreds of people, grounding all 737 MAX aircrafts around the world, and costing the company billions. While investigators can point to a malfunctioning part in the airplane, the hundreds or thousands of decisions and actions that ultimately produced that problem are far more important.

Disaster incubation is how some safety experts describe the above stories. Over time, organizations become complacent about safety and then, eventually, something awful happens. Dekker wrote:

*Drifting into failure is a gradual, incremental decline into disaster driven by environmental pressure, unruly technology, and social processes that normalize growing risk. No organization is exempt from drifting into failure.*¹

So, what can you do to minimize drifting into failure? Become more resilient. Dekker explains, "Resilience represents the ability to recognize, adapt to, and absorb problem disturbances without noticeable or consequential decrements in performance."² High-reliability organizations (HROs) are resilient. What drives that resilience? Diversity.

According to Dekker, diversity enables "a larger number of perspectives to

view a problem with and a larger repertoire of possible responses.”³ Greater diversity (be it through race, gender, age, educational rank, or status) leads to better problem solving, which produces greater resilience. This assumes that diverse voices in the organization are empowered and encouraged to speak up and that everyone doesn’t just listen to the most senior person in the room. Dekker elaborated:

*By encouraging diversity, HROs show a considerable decentralization of decision-making authority about safety issues. This permits rapid and appropriate responses to dangers by the people closest to the problems at hand. Wildavsky called this “decentralized anticipation,” which emphasizes the superiority of entrepreneurial efforts to improve safety over centralized and restrictive top-down policies or structures.*⁴

For example, Dekker describes how “even the lowest ranking individual on the deck of an aircraft carrier has the authority (and indeed the duty) to immediately suspend any takeoff or landing that might pose unnecessary risk.”⁵

Becoming a genuine HRO isn’t easy. It requires creating a culture that rewards and encourages dissent and speaking up (psychological safety), empowers lower-ranking employees (such as nurses in hospitals), grows diversity within the organization, decentralizes safety decisions, and encourages entrepreneurial problem solving. Creating such a culture increases your chances of becoming a high-reliability organization.

Harness the power of diversity and decentralized control to make safety a prerequisite and avoid drifting into failure.

¹. Sydney Dekker, *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems* (Boca Raton, FL: CRC Press, 2011).

². Ibid.

³. Ibid.

⁴. Ibid.

⁵. Ibid.



Stop Work Authority

Fixing problems promptly was key to producing quality vehicles at Toyota, which was essential to the company's success. Therefore, work was designed to allow workers to find and fix manufacturing problems long before they became bigger, more expensive problems to solve. Toyota's assembly line workers could pull a rope (called an "Andon" Cord) to halt production whenever they spotted a quality issue. Once the cord was pulled, a group would gather to discuss and solve the problem. Workers didn't just have the freedom to pull the Andon Cord; it was considered their obligation.

Stop Work Authority (SWA) cards are common in safety-conscious industries like manufacturing, oil and gas, electric, and nuclear power. Everyone is given an SWA card, and like an Andon Cord, they are empowered to use it to halt any unsafe work.

I once attended a workshop about safety culture and was the only software person in the room. Everyone had an SWA card and I had no idea what they were. Once I learned about them, however, I wondered why other industries didn't have them. Surely my own industry, tech, is hazardous and capable of injuring people and the companies they work for.

After the workshop, I immediately created SWA cards and gave them to all our employees.

After everyone received their SWA cards, we had a company-wide discussion about how the cards could help us recognize and remove injuries, hazards, and near misses related to our time, money, information, reputation, relationships, and health.

A few weeks later, an employee was extremely jet-lagged, yet still attempting to work. Another employee, with whom he was collaborating online, recognized how exhausted and ineffective he was. That employee flashed her SWA card and our tired traveler went home to get some much-needed rest.



Another time, an employee had a severe toothache, yet he was working through the pain—attempting to prepare new materials for a business trip the following week. Again, a colleague stepped in, flashed the SWA card, and worked with others to find a replacement for the work and business trip. The employee with the toothache learned that he needed an emergency root canal.

On yet another occasion, an employee flashed his SWA card at *me*. A new marketing manager was organizing a public workshop and it was her first time doing the work. We didn't have a good description of how to plan these

workshops back then, so the work was hazardous.

After years of experience, I'd learned not to get locked into scheduling an expensive hotel conference room for a public workshop too early, since you can lose a lot of money if you must change workshop dates. But our new marketing manager didn't know that. I was in a rush to leave the office to catch a flight, and I quickly sent out an email response to our marketing manager about hotel conference room bookings. My words weren't exactly patient or polite, and a colleague who was on the email thread noticed that I wasn't communicating with the utmost dignity and respect. So he flashed the SWA card. It was a good lesson for me.

SWA cards empower your staff to help create a safer workplace.

Empower everyone to help make safety a prerequisite.



Become Resilient

When I founded Industrial Logic in 1996, building a successful business was topmost in my mind. During the preceding seven years, I'd been designing and developing custom risk-management software for Bankers Trust, a Wall Street bank. I loved the work and spent lots of time learning how to become a better software designer. Managers valued my work and my income rose steadily. By 1995, however, I was ready to give up this lucrative position to start my own company. The entrepreneurial urge I was feeling wouldn't let me rest until I did.

I had squirreled away some modest savings and I was pretty certain there were companies that could use my help. I founded Industrial Logic in early January 1996, based in my tiny studio apartment in Manhattan. I had no income, no clients, and little knowledge of how to actually run a company.

"Most businesses fail" was one of the sentences I read in *Entrepreneurship for the Nineties*, a guide to starting up and running a business, written by Gordon Baty. I read many such books at the time because I was determined to succeed, and I hoped their collective wisdom would rub off on me. I worked hard and learned how to live on little income.

By the late 1990s, I had managed to build a small client roster, but my business was still fragile. Some of the more established consultants I knew suggested I read *The Secrets of Consulting*, by Gerald M. Weinberg. In retrospect, it was this book that had the greatest impact on my success. One of the most important secrets I learned from Weinberg's book was how to make my business resilient.

When you start a company, you usually have zero clients and are hungry for work. After you win your first client, it's tempting to give that client *all* of your time, especially if they ask for it. Weinberg cautioned against this. He said, "Never let a single client have more than one-fourth of your business." Why? Because an "unhealthy dominance by one client" makes your business vulnerable. When the work with that one client ends, which it inevitably will, you have no other clients or income to fall back on and it takes time to sign new ones. It's just like getting unexpectedly fired from a full-time job—you're suddenly scrambling to find a new job to fill the gap.

Weinberg told the cautionary tale of his friend Wesley, a consultant who had multiple clients taking up about 50 percent of his time. When one of those clients asked Wesley to go full-time, he succumbed to the temptation to make more money. After his two-year contract with that customer ended, Wesley had

no new customers lined up and he was forced to shut down his consulting business and find a full-time job.

Another one of Weinberg's invaluable secrets related to marketing. He said, "Spend at least one day a week getting exposure"—cautioning against being so busy that you neglect promotional activities. He suggested regularly publishing your writing, giving talks, or conducting seminars. Such exposure could lead to your next client while building resilience.

Weinberg's wisdom helped me grow a thriving business, built on a solid foundation of resilience.

Become resilient. It's a secret to surviving and thriving.



Summary: Drive Out Fear

Safety is a doorway to excellence. It enables confidence, and confidence enables speed. Invest in protecting people by making safety a prerequisite. Learn about and eliminate anything that makes customers unsafe. Increase safety rather than blaming or penalizing people. Look for risks and remove them safely. What are your brown M&M's?

High-performing teams operate within a psychologically safe climate that allows people to be comfortable discussing and reporting errors, exploring ways to remove errors, and continuously improving. How safe is it to fail in your world? How could you test it?

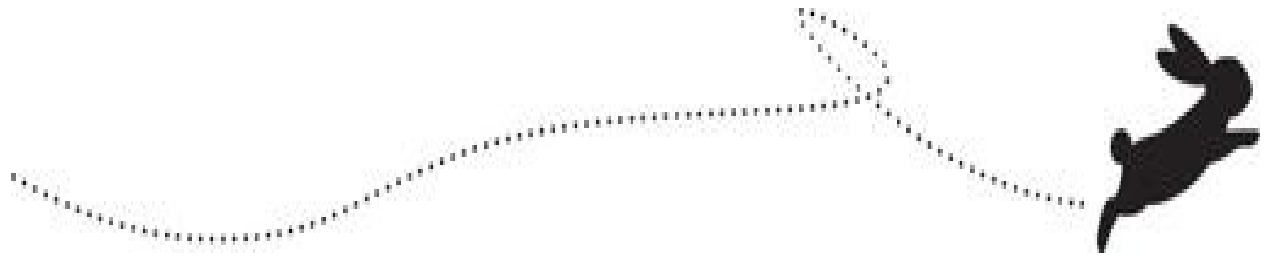
In your own work with colleagues, be keenly aware of what you protect for yourself and what you protect for the larger group. Spartan warriors made sure to protect the larger group by never losing their shields.

Safety is not a miracle, it's a choice. Replace heroics with agility. Heroism is hazardous. Don't leave safety to chance. Proactively analyze hazards and handle them before they harm you. Just because something works doesn't mean it will continue to work.

Healthy fear has its place. Pay attention to it. Be careful (but don't play it safe). Think hard before remaining silent. Your voice matters.

Resolve conflict quickly. Effective collaboration requires trust. Work hard to identify and break cycles of mistrust. Use an anonymous safety poll to learn whether it's safe to speak up, regardless of who's in the room.

Harness the power of diversity and decentralized control to make safety a prerequisite, and avoid drifting into failure. Empower everyone to make it happen. Becoming resilient is a secret to surviving and thriving.



Be Readily Resourceful

The word *readily* means “without hesitation.” Agility involves being readily resourceful when facing challenges. People who are resourceful have an ability to “find quick and clever ways to overcome difficulties.” Being readily resourceful means not hesitating to overcome challenges, maneuver around obstacles, and quickly solve problems.

That isn’t always an easy thing to do.

Many of us are resentful of unforeseen obstacles or difficulties. Sometimes they make us feel helpless, anxious, or stuck. Those feelings are a good indicator that it’s time to become readily resourceful. Doing that requires an attitude that obstacles and challenges can be overcome, that they are meant to be overcome. Being stuck or helpless is by no means a permanent condition. You can be clever, consider new ideas, learn from others, and discover unexpected solutions.

Being readily resourceful requires optimism as well. Treat obstacles and challenges like puzzles to solve. There’s *always* a solution. Consider options, leverage diverse points of view, ask many questions, and determine whether constraints are real or imaginary.

Be suspicious of the status quo. Challenge conventional wisdom. Defy assumptions.

Spark change. Embrace change. Be change.



Birth of an Airline

Sometime in the late 1970s, Richard Branson and his girlfriend were stranded in the Virgin Islands when their flight to Puerto Rico was abruptly canceled. There were no weather issues, and the airline gave passengers no indication of when another flight would depart.

Instead of remaining stuck in the airport and waiting around for a new departure time, Branson was readily resourceful. He called several charter flight companies, found one that would fly from the Virgin Islands to Puerto Rico for \$2,000, and booked it. He then divided the price to charter the airplane by the number of seats and made a sign that read: “VIRGIN AIRWAYS: \$39 SINGLE FLIGHT TO PUERTO RICO.” Holding the sign as he walked around the terminal, Branson quickly filled every seat.



The group flew uneventfully to Puerto Rico and Branson had the spark of an idea for what would one day become Virgin Atlantic Airways.

An unexpected event is an invitation to be readily resourceful.



A Nordstrom Ninja and a Raving Fan

Two weeks before her wedding, my friend Sandra saw a display of custom necklaces at Nordstrom. The necklaces were locally made to order using Swarovski crystals and looked absolutely stunning. Sandra was assured by Michelle, a Nordstrom salesperson, that the custom necklace Sandra designed would be delivered to the store in time for her wedding. So she placed the order.

Two days before her wedding, Sandra went to pick up the necklace, but it wasn't what she had commissioned. She decided not to accept it. Deeply disappointed, Sandra told Michelle that she would just borrow a necklace from a friend.

Michelle wasn't ready to concede defeat just yet. Unbeknownst to Sandra, Michelle went into full ninja mode. She drove to the factory that made the necklaces, asked them to immediately create the correct custom jewelry, beautifully wrapped the piece, and then drove an hour to Sandra's hotel to personally deliver the necklace into Sandra's hands.

"Shocked" and "amazed" are the words Sandra uses to describe how it felt to know that someone cared so much to help her. The necklace was the perfect complement to her dress, and she wore it with matching earrings on her big day. Sandra still has the necklace, and every time she sees it, she remembers Michelle and Nordstrom's commitment to her happiness.

Books and articles have celebrated the "Nordstrom Way" and the autonomy given to sales associates to exceed customer expectations. Internally, Nordstrom celebrates stories of exceptional customer service on its website and internal postings. Many people who regularly shop at the store have their own Nordstrom story.

Sandra's Nordstrom story resulted from Michelle being quick, resourceful, and adaptable in solving the necklace problem and delighting her customer. What enabled such agility? A culture that genuinely values customer happiness and empowers employees to deliver it however they see fit.

In his book *Raving Fans*, Ken Blanchard explains why and how to create raving fans rather than merely service customers. I learned about Sandra's story during our company's study group on Blanchard's book, and now I know why she is a raving Nordstrom fan.

Empower your staff to be resourceful so they may exceed customer expectations.



Readily Resourceful in a Recession

By 2008, my company had been training and coaching Google engineers in agile software design for four years. Our workshops were so popular that we had our own classroom on the Google campus in Mountain View, California. We taught classes several times per month, and there was a long waiting list to get into our workshops.

Back then, Google was our cash cow, so it was alarming when signs of a recession sprang up and Google moved quickly to tell *all* of its vendors that they would no longer be able to pay for services. Hundreds of thousands of dollars in revenue replaced by nothing, almost overnight? That would be challenging for our business. Meanwhile, based on a steady stream of positive student evaluations, Google was extremely happy with our services.

Google's training division knew that its engineers still needed to learn what we taught. Sometime in 2008, one of Google's senior engineers, a guy named Chris, asked to learn how we taught our workshops so he could help teach them, too. Since there was a long list of students waiting to get into our workshops, we agreed to let him become an instructor provided that we could ensure he was qualified to teach and that we could collect a small fee when he taught. Chris observed me teaching the workshop and then taught it himself with me watching him. He did well and was permitted to teach our workshops.

By then, the global recession was threatening to completely eliminate our business at Google. Chris volunteered to teach all of our workshops, since Google was no longer allowed to pay for any external services. We'd make a small fee, but it would be a pittance compared with what we'd been previously making there. Meanwhile, there were thousands of Google engineers in offices around the globe who needed our training, and one instructor could not handle that load.

One important part of our workshops was that we relied on our own eLearning materials in class. When we started teaching at Google in 2004, we had developed and used our eLearning software, including videos, quizzes, and exercises, to help engineers learn important skills. Now that Google was no longer permitted to pay for services, we suggested that they just buy our eLearning courses and use them to help educate engineers, since Google was still allowed to buy software.

We had a meeting with Google to discuss the idea, but the individuals

weren't interested. The problem, they said, was that while our eLearning was good, it could not replace the invaluable experience of having an instructor walk around the classroom and give feedback to students as they worked on programming exercises. Chris made this argument and lobbied to teach *all* of our workshops, even at Google offices around the world.

We were not ready to concede.

Back in our office, we wondered how our eLearning software could “look over students’ shoulders” and automatically provide feedback when they worked on one of our challenging programming exercises. We began a series of discoveries and experiments. We learned that the tool used by programmers to write code already kept a history of some of the data we needed to automatically provide feedback to students. What it lacked was a history of when programmers ran tests. We quickly discovered a way to obtain that data. Now, using all of this historical data, we designed a way to automate feedback to students. It was crude at the time, but it worked!

I requested one more meeting with Google’s director of education. In a small conference room in Mountain View, I demonstrated our new “automated critique” and how it could provide feedback to students without us being in the room. I’ll never forget the director’s response. She said, “This is a game changer!” I walked out of the meeting elated. A few months later, we sold \$600,000 of our eLearning licenses to Google!

That income helped us weather the global recession without letting anyone go or reducing salaries. Meanwhile, we refined our software to automatically offer precise feedback on student exercises, a level of detail that no human instructor could possibly provide to every student. Automated critiques became our “killer feature” and helped us sell our eLearning to many clients for years to come.

It’s true that “necessity is the mother of invention.” The global recession was an unexpected event that required my company to be resourceful (“able to find quick and clever ways to overcome difficulties”). We rose to the challenge. This experience ended up being one of the most exciting and productive times in my career.

The right resourceful response to an unexpected event could be
your game changer.



The Kondo Way

Before Marie Kondo wrote her best-selling book *The Life-Changing Magic of Tidying Up*, she learned from another author not to seek perfection in decluttering her home. “Start off slowly and discard one item per day,” wrote that author. Marie decided to follow this sensible advice, reasoning that aiming for perfection from the beginning is daunting, and perfection itself is unattainable.

Before two weeks were up, Marie abandoned this advice. Her home was still cluttered, and she realized that she was accumulating things faster than she was discarding them. She also realized that the little-by-little style of decluttering didn’t match her personality: “I’m not the type of person who likes to plug away at something, one step at a time,” she noted.

An agile coach in a South Pacific bank once asked me how to keep her energy and enthusiasm up given that the bank was already two and a half years into their agile transformation. I could tell that she was frustrated by the slow pace of change. It reminded me of Marie Kondo’s little-by-little experiment. I shared that, in my experience, companies take years to change and that two and a half years wasn’t so long compared with other organizations. I suggested she exercise patience. But I didn’t like my answer.

I’m more like Marie. Slogging through change, one little step at a time, is too slow and ineffective for me. I prefer change that is faster and more meaningful, the kind of change that produces better outcomes. The companies I visit around the globe are in serious pain from ineffective teamwork, insufficient risk management, money and power struggles, and inadequate leadership. Little-by-little change runs the risk of barely addressing these problems.

To really change mindsets and not revert to older ways of working, people need to experience deep and meaningful results within a relatively short period of time. Marie Kondo said:

*When people revert to clutter no matter how much they tidy, it is not their room or their belongings but their way of thinking that is at fault. Even if they are initially inspired, they can’t stay motivated and their efforts peter out. The root cause lies in the fact that they can’t see the results or feel the effects. This is precisely why success depends on experiencing tangible results immediately. If you use the right method and concentrate your efforts on eliminating clutter thoroughly and completely within a short span of time, you’ll see instant results that will empower you to keep your space in order ever after.*¹

So how do you achieve such deep and meaningful results within a relatively short time? Pick something important and then go deep to make it a success.

In the professional world, that typically involves assembling the right multidisciplinary team, clearly identifying the strategically important goal for the team to achieve, training and coaching the team in high-performance skills, and providing the support and environment the team needs to succeed.

This all-at-once approach isn't for everyone and isn't always the right choice. For example, if you are new to an organization, you may not have the political capital to go after something big. In that case, a slow-and-steady approach may be superior.

When changes aren't happening soon enough and a team or organization is suffering, then "going big" can really help. I've seen it lead to deep, high-impact, I'll-never-go-back-to-the-old-way change that sets in motion even more meaningful changes. People become more joyful at work, they are proud of what they're doing, and customers are delighted.

All-at-once change may seem daunting until you actually try it and experience the results. In general, I have found it to be far more effective than slow, little-by-little change.

If a little-by-little approach doesn't make much of a difference,
try the life-changing magic of going big.

¹. Marie Kondō, *The Life-Changing Magic of Tidying Up: The Japanese Art of Decluttering and Organizing* (Berkeley: Ten Speed Press, 2014).



Bullet-Train Thinking

I love the stories in Charles Duhigg's excellent book *Smarter Faster Better: The Transformative Power of Real Productivity*. Two of those stories exemplify what it means to be resourceful, and I retell them frequently.

After the devastation of World War II, Japan desperately needed to get back on its economic feet. One of the first areas to improve was the country's railway system, which was habitually slow. To quickly rebuild the economy, Japan needed to be able to move people and resources around the country much faster, and railway engineers were given the task to dramatically speed up the trains. They assumed that meant designing trains that could travel at around sixty-five miles per hour, which would've been the fastest trains in the world at that time (the mid-1950s). Actually, the head of the Japanese railway wanted the trains to go much faster than that.

How fast? More like 120 miles per hour, a crazy, unheard-of speed at the time.

The railway engineers complained that a train traveling at 120 miles per hour would derail on a curve in the tracks due to centrifugal force. The railway head asked why there needed to be curves in the tracks—the engineers were empowered to change the *entire* railway system. And they did, slowly and methodically, with experiment after experiment, over a period of about nine years. Some of these experiments, like putting an engine on every train car, were failures. Others showed promise. This awesome goal gave railway engineers the autonomy to question everything and try lots of potential solutions. Duhigg explained the phenomenal result:

In 1964, the Tōkaidō Shinkansen, the world's first bullet train, left Tokyo along continuously welded rails that passed through tunnels cut into Japan's mountains. It completed its inaugural trip in three hours and fifty-eight minutes, at an average speed of 120 miles per hour.

Several decades later, Jack Welch, CEO of industrial giant General Electric (GE), visited Japan and learned how Japan's railway engineers invented the bullet train. He immediately saw a need for "bullet-train thinking" at GE. When he returned from his trip, he asked every division within GE to present a major stretch goal.

One such division was airplane engines, operating out of Durham, North Carolina. That division came back to Jack with the stretch goal to cut defects in engines by 25 percent. That wasn't audacious enough (in fact, the division already mostly knew how to succeed at that goal). Jack countered, "Reduce 70

percent of defects within three years.” *That* was bullet-train thinking!

Managers analyzed errors and quickly realized that more of their staff needed to be trained in quality assurance. Everyone got training but that didn’t yield enough improvements. They then tried hiring more engineers who knew their way around the ten thousand parts inside a jet engine. But there was a great deal of competition to hire such people. So the division offered them greater autonomy, less centralized planning, and more self-organization. That hiring approach worked and more experts joined the division and made changes to it.

Within three years, the jet engine division had thoroughly transformed and succeeded far beyond its wildest expectations. Duhigg wrote:

*By the time they were done, the Durham plant’s managers had collapsed organizational charts, remade job duties, and overhauled how they interviewed candidates, because they needed people with better team skills and more flexible mindsets. In other words, Welch’s stretch goal set off a chain reaction that remade how engines were manufactured in ways no one had imagined.*¹

In my decades of experience helping companies in many industries, I’ve seen a lot of organizations stagnate. Whether it’s due to poor leadership, fear of risk taking, fear of being blamed, politics, infighting, acceptance of the status quo, or some combination thereof, the problem is common.

James Womack, the father of the lean movement (and author of numerous books, including *The Machine That Changed the World* and *Lean Thinking*), observed within the lean community a “growing quietism—an acceptance of things as they are without attempt to resist or change them.” He described a shift from “lean leaps for dramatic organizational transformation” to small-scale improvements that fail to yield significant improvements.

This is why I love bullet-train thinking. It directs us to envision something truly awesome and then enable and empower people to achieve it, even if that means rethinking everything. Bullet-train thinking is enormously effective in helping people become resourceful and stretch to achieve what was once considered impossible.

Unleash resourcefulness and achieve awesome outcomes via
bullet-train thinking.

¹. Charles Duhigg, *Smarter Faster Better: The Transformative Power of Real Productivity* (New York: Random House, 2016).



Questions Are the Answer

When he became Alcoa's chief information officer, William O'Rourke knew next to nothing about computers. "About all I knew back then," he said, "was how to reboot a computer using the control-alt-delete keys."

O'Rourke inherited an IT organization with 108 data centers globally. He knew he wanted to help make the IT organization excellent, but he didn't have a road map for improvements. So he began by asking questions.

"How many data centers would the best company in the world have?" he asked his staff. They told him that the best would have just two data centers, located on different continents and capable of backing each other up.

"What obstacles are preventing us from doing that?" he asked next. This question helped William's staff understand that he expected them to question the status quo. His staff began to document the obstacles, and that helped them create a plan of action.

Another question was, "What do we do with all of the extra hardware?" The 108 data centers scattered around the world contained a lot of computers, most of which would not be needed when the number of data centers was dramatically reduced. The team asked IBM how they could help. IBM agreed to take back Alcoa's old equipment if they were awarded contracts for the new equipment.

Rather than telling people what to do, William kept asking questions, with a focus on empowering his team to create a world-class IT organization. In just under two years, there were only four data centers for the entire global company. It was a huge reduction from 108, saving Alcoa millions of dollars.

In his book *Questions Are the Answer*, MIT professor Hal Gregersen explains how successful leaders ask the right questions. Such questions inspire people to challenge the status quo, see things with fresh eyes, and foster a creative problem-solving environment, much like William O'Rourke's IT organization.

How do you inspire people to make big improvements? Questions are the answer.



The Wright Stuff

Wilbur and Orville Wright felt defeated following their 1901 airplane experiments. While in the air, Wilbur described “a peculiar feeling of instability,” which sometimes became genuine fear when he discovered that their craft was seriously unpredictable. Twice it climbed rapidly out of control before stalling, and once the machine began falling backward. The craft’s rudder helped Wilbur regain control, but the unexpected movement was sufficiently frightening that the brothers decided “to pause and gather a full range of data while flying the glider as an unmanned kite before risking any further damage to craft or pilot.” On the train ride back to Dayton, Ohio, Wilbur said to Orville, “Not within a thousand years would man ever fly!”

He was wrong by 998 years.

On December 17, 1903, the Wright brothers flew the world’s first airplane 120 feet through the air, landing safely twelve seconds later. Later that day, they flew 852 feet in fifty-nine seconds. They had finally designed an aircraft that could be controlled in three axes of motion (called pitch, roll, and yaw), something humans had never successfully done before.

Why had they succeeded when so many before them had failed?

The brothers knew that the problem of human flight “could not be solved by stumbling upon a secret, but by the patient accumulation of information upon a hundred different points.” To uncover that information, the brothers would need to be more resourceful and to embrace change better than anyone who had come before them. And what would enable them to do that was their superb mechanical skills, strategic scholarship, scientific precision, safe experimentation, and steadfast determination.

The brothers inherited their mechanical skill from their mother, Susan. Raised on a farm in Indiana, she became adept in woodworking and metalworking by helping with her father’s carriage construction business. As a wife and mother, Susan put her considerable mechanical skills to work crafting toys for the kids and designing simple home appliances. Her children learned from their mother and came to her when they needed mechanical advice. Orville built and sold his own kites, and Wilbur invented a paper-folding machine after tiring of manually folding a weekly batch of church newspapers. As young adults, the brothers built and operated their own printing press and later repaired and manufactured bicycles.

The Wright brothers were scholars. Orville once said, “We were lucky enough to grow up in an environment where there was always much

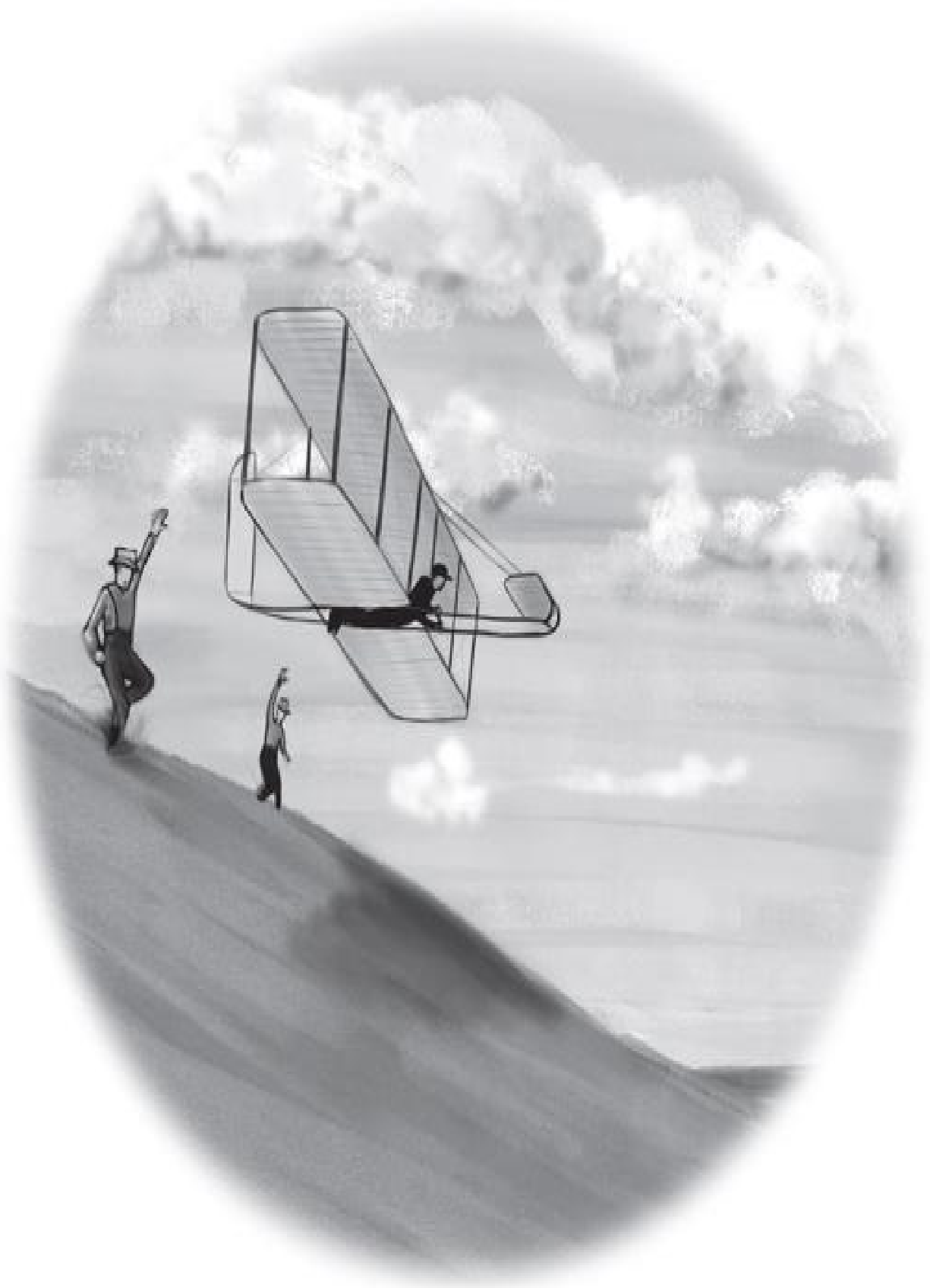
encouragement to children to pursue intellectual interests; to investigate whatever aroused curiosity.” Susan and Milton, her husband, attended the same college and filled their home with books. After a devastating hockey injury prevented Wilbur from attending Yale University, the family’s bookshelf sustained his intellectual curiosity during the ensuing years.

In 1899, Wilbur began a strategic study of the existing literature on flight, which Wright biographer Tom Crouch described as “a morass of conflicting opinion, speculation, and guesswork that passed for the literature of aeronautics.” While others had spent decades making sense of this literature and writing books about their ideas, Wilbur arrived at an accurate understanding of the state of the art in only a few months.

Orville and Wilbur preferred scientific precision over guesswork in every aspect of their aviation experiments and research. Using Otto Lilienthal’s formula for calculating lift and Octave Chanute’s formula for calculating drag, they determined that a glider weighing 190 pounds, with a surface area of two hundred square feet, would need a fifteen-mile-per-hour wind in order to fly. Instead of guessing where to find such wind conditions, they requested data from the U.S. Weather Bureau in Washington and received average hourly wind velocities recorded at 120 Weather Bureau stations. That data helped them select Kitty Hawk, North Carolina—a place they’d never heard of—as an ideal location for their flying experiments.

The Wright brothers approached the danger of aviation with meticulous safety. Before Wilbur and Orville began their first flying experiments at Kitty Hawk in 1900, their widower father, Milton, was concerned for their safety. To assuage Milton’s concerns, Wilbur explained how he would avoid danger:

*In my experiments I do not expect to rise many feet from the ground and in case I am upset there is nothing but soft sand to strike on. I do not intend to take dangerous chances, both because I have no wish to get hurt and because a fall would stop my experimenting, which I would not like at all. The man who wishes to keep at the problem long enough to really learn something positively must not take dangerous risks. Carelessness and overconfidence are usually more dangerous than deliberately accepted risks.*¹



One of the biggest problems the Wright brothers encountered was what they believed were errors in the lift and drag data they had inherited from Lilienthal and Chanute. They embraced change and delved deeply into solving the problem. Using their superb mechanical skills, they constructed a wind tunnel out of wood, wire, glass, hacksaw blades, and a fan, then used this delicate device to test tiny fluctuations in lift and drag on hundreds of surfaces and shapes set at various angles to the wind. This groundbreaking work revealed the correct coefficient for air pressure and uncovered answers to previously unanswered questions, such as “What is the ideal distance between the two wings in a biplane?” and “Should the upper and lower wings in a biplane have the same camber?” The wind tunnel helped the Wrights produce data they could trust and led them to the ultimate breakthrough of a practical flying machine.

Others who came before the Wright brothers were surely resourceful and embraced change. What set the Wright brothers apart was their effective mix of mechanical skills, strategic scholarship, scientific precision, safe experimentation, and steadfast determination. It was the Wright stuff for inventing the world’s first airplane.

To maximize your chances of achieving an outstanding outcome,
make sure you have an effective mix of deep skills.

¹. Tom Crouch, *The Bishop’s Boys: A Life of Wilbur and Orville Wright* (New York: W.W. Norton & Company, 2003).



It Was All Greek to Me

Have you ever struggled to learn something and then decided you just weren't cut out to learn it? Harboring such self-limiting thoughts can hold you back from achieving your potential at school, work, or home. I fell into this common trap in childhood and managed to escape it, thanks to a transformative experience that took place during the summer after my freshman year of college.

In middle school and high school, I studied French. I liked the language, yet I was not a great student. Even though I paid attention in class and completed my homework, I wasn't good at memorizing vocabulary and verbs. My grades suffered, and I was convinced that I wasn't good at memorizing and therefore wasn't good at learning foreign languages.

After graduating from high school, I enrolled in St. John's College, a pure liberal arts school. This school was the definition of "old school"—I was required to study ancient Greek for the first two years. I decided to do whatever it took not to repeat my bad experience with French.

I studied hard my first year, but it was like running through quicksand. No matter how hard I tried, I could not translate ancient Greek effectively. What I didn't know at the time was that the Greek textbook we were using was actually making the study of ancient Greek more difficult than it needed to be.

That freshman year I studied *all* of the time, even on my winter break between first and second semester. But when I returned for the second semester, my translation abilities had not improved one iota. I was extremely frustrated but refused to give up.

A few weeks before the end of my freshman year, I happened to spot a brochure pinned to a wall near a water fountain on campus. It was from the Latin/Greek Institute of the City University of New York and described an intensive course in ancient Greek offered during the summer. The course covered four to six semesters of college-level Greek in only ten weeks.

I was sold.

The Institute began in early June. On the first day of class, every student from the Latin and Greek programs met in an auditorium. The instructors explained what lay ahead and were brutally honest, telling us that half the people in the auditorium wouldn't finish the program.

Classes went into full swing on day two. The founder of the Institute, Dr. Floyd J. Moreland, strode into our class, introduced himself, and launched into a lesson on the ancient Greek verb system. Ten minutes into his lecture, he recited all six principal tenses of the verb *luo*, meaning "to loosen or unbind":

λύω	λύσω	ἔλυσα	ἔλυκα	λέλυμαι	ἐλύθην
luo	lusso	elusa	leluka	lelumai	eluthain

He asked us to repeat the tenses of this verb with him. We stumbled through it. I sat—amazed, awed, and stunned. Here, just two days into the course, we weren't just being asked to learn the simple present and past tenses of a verb. We were being asked to learn every single tense, every conceivable way of saying the Greek verb “to loosen or unbind”: “I am loosening,” “we will loosen,” “she had loosened,” “they were loosening,” “he may loosen,” “you might loosen,” and so on. For one verb alone, that meant memorizing six tenses, each of which had six persons (“I,” “you,” “he/she/it,” etc.). That's thirty-six words to memorize for just one verb!

Every evening, we were given numerous verbs and other vocabulary to memorize. Many of my classmates left the program after the first week. I met with one of my instructors to see if he had any suggestions for how I could keep up with this fire hose of information. He told me to start making index cards and carry them around with me. I told him that index cards didn't work for me—I had tried using them before to learn French, and they didn't help.

“Make the index cards,” he told me.

So I made the cards, and I carried them with me everywhere.

We had daily quizzes, and the days were sixteen-plus hours of lectures and translations, reciting Greek, reading it, listening to it, and writing it. It was total immersion. A few weeks into the program, the teachers began calling us “hoplites,” the Greek word for soldiers who fight at the front of battle lines.

As the weeks passed, more students dropped out. I struggled to keep up, and my social life was nonexistent. Each morning, I memorized verbs and vocabulary every possible moment in every place possible until I got to the classroom: waiting for the bus, on the bus, waiting for the train, on the train, walking to school, riding the elevator, walking to class. I vividly remember the Fourth of July. I had been invited to go sailing to see the fireworks. Yet, as usual, I had a big Greek test to study for. So my index cards accompanied me on the boat. They followed me *everywhere*.

Then it happened.

I noticed it was getting easier for me to memorize the verbs. While it had been difficult to memorize thirty-six ways to use the verb *luo*, it turned out that the six principal parts of the verb “to learn” sounded just like the endings of the six principal parts of *luo*. There were patterns and they made it far easier to learn more and more words without a lot of effort.

In hindsight, I know now that the traditional approach to teaching and

learning a verb would never have revealed these patterns since, traditionally, only one tense is taught at a time, with many weeks separating the study of each new tense.

In addition to the patterns, the Greek textbook itself made learning easier. *Greek: An Intensive Course*, written by two of the Institute's instructors, was a brilliant book. Every lesson was crafted to not only introduce new material (grammar, vocabulary, verbs), but also to offer a continuous and thorough review. My amazement at just how good this book was versus just how bad the textbook I'd been using at college was made me realize how important a good book can be. It was simply remarkable for me to consider how big a difference there is in outcomes when you use a good book instead of a bad book. To this day, I constantly seek out the clearest and most intelligent treatment of a subject.

As time went by, it was exciting to realize that my memory was improving. I could just look at sentences and remember them without looking at my book! There was nothing deficient about my ability to memorize—it had been there all along. I just needed to give it a good workout.

The Latin/Greek Institute changed how I viewed myself. I discovered that I had the ability to do what I had convinced myself I could not do. When I returned to college for my sophomore year, I was far more confident about my Greek, to say the least. I quickly realized that I knew more than my Greek professor, and I would frequently correct her in class. This was so odd, given how bad I had been at Greek the previous year. I enjoyed my newfound ability, but I didn't want others to spend a year as I had, struggling so hard and hitting so many walls.

So, I did two things: I started tutoring freshmen who were having difficulty learning ancient Greek, and I talked with the dean about replacing the old textbook with the one I had used at the Institute. I explained to him how much I had struggled the previous year, and how I was now far ahead of my fellow sophomores and even some of my professors—primarily because I learned Greek through a much more intelligent process. I made a very strong case for getting rid of, or rewriting, the school's textbook, which a few of the school's professors had written many years back.

And to my delight and amazement, the dean took action. By the time my senior year began, freshmen were studying from a completely new Greek textbook, based on the method used by the Latin/Greek Institute.

This transformational educational experience was a turning point in my life. My self-esteem skyrocketed as I no longer harbored self-limiting thoughts about my learning abilities. From that point forward, including into my professional career, I would take responsibility for my learning and not let obstacles limit my

potential. This was super important as I taught myself software design. If I was struggling too much to learn a software concept or technique, I'd look for another book or explanation rather than fault myself. It was liberating and productive to have this outlook on my education.

Stop faulting yourself if you struggle to learn something. Instead, be resourceful by seeking a better explanation, book, teacher, or process.



Take Personal Responsibility for Learning

One night, in a Dallas hotel room, I couldn't sleep because the toilet wouldn't stop running. It was too late to call the front desk and ask for maintenance to fix it. But I was scheduled to teach a workshop in the morning and needed to sleep!

What was wrong with this hotel?!? Couldn't the staff at least take care of the rooms? Why was I paying good money to have my sleep disturbed? I imagined the conversation I would have with the front desk in the morning. Maybe I would demand a refund or discount, or at least a sincere apology for my inconvenience. My head was filled with thoughts of blame. Somehow, through it all, I managed to fall asleep.

In the morning, the toilet was still running. I went into the bathroom and decided to jiggle the handle. Within a few seconds, the toilet stopped running. OMG! I could have fixed this problem myself—I could have gotten a good night's sleep!

At the start of the workshop that morning, my colleague Ashley Johnson introduced the students to a model created by Christopher Avery and Bill McCarley called The Responsibility Process:¹

Ashley explained that during the workshop, students might find themselves in one of the mental states described in the model. For example, a student might *lay blame* on another student for messing something up during a group exercise. Or a student could learn something new, but then *justify* that nothing will ever change in their organization.

Responsibility

Owning your ability and power to create, choose, and attract

OBLIGATION

Doing what you should or have to instead of what you want to

Giving up to avoid the pain of Shame and burden of Obligation

QUIT

SHAME

Laying blame onto oneself, which is often felt as guilt

JUSTIFY

Using excuses for things being the way they are

LAY BLAME

Holding others at fault for causing something

Ignoring the existence of something

DENIAL

THE RESPONSIBILITY PROCESS®

The Responsibility Company™  Learn more at Responsibility.com

Ashley asked the students if they would like to take *responsibility* for their learning. What did that mean? For example, instead of waiting until the end of the workshop to blame the instructors for not doing or teaching something, students could take responsibility to ask the instructors for what they needed *during* the workshop. This is when I told the students about my experience with the running toilet and how I could have taken responsibility to try to fix the problem myself rather than dwelling in a state of blaming others for my predicament.

All of the students in our workshop then agreed to take responsibility for their learning.

When years earlier I was struggling to learn French and ancient Greek in school, I didn't know it, but I was working my way through The Responsibility Process. While I didn't *lay blame* on others for not being good at memorization, I did blame myself for not working hard enough.

Then I *justified* my lack of talent at memorization by deciding that knowledge and wisdom had nothing to do whatsoever with being able to memorize facts or figures.

Then I felt *shame* about the low grades I received in French and about struggling to learn ancient Greek during my freshman year of college.

I considered *quitting* French, but I didn't want to give up.

Then I felt an *obligation* to continue my studies despite the poor results I was getting.

Finally, after discovering that I had no problem with memorization and that a far better method existed for learning ancient Greek, I understood that I could play a far more powerful role in my learning. If a book, course, or teacher was making it hard for me to learn a subject, I could find a *better* book, a *better* course, or a *better* teacher.

I wasn't stuck. I could take *responsibility* for my learning!

It's easy and natural to fall into mental states that avoid taking responsibility. Christopher Avery and Bill McCarley explain that learning only happens when we are in a state of responsibility. In that state, we can overcome almost any challenge and grow in understanding.

Take responsibility for your learning. Jiggle the handle.

¹. Christopher Avery, *The Responsibility Process: Unlocking Your Natural Ability to Live and Lead with Power* (Partnerworks, Inc., 2016).



Business as Usual

Willem Larsen, a software development and process coach, was shocked and disappointed. After a team he had coached in cutting-edge software development practices produced excellent results, he assumed that the company's management team would want to leverage what had worked so well. But nothing could have been further from the truth.

Months earlier, senior leaders in a large automotive company described an extremely important piece of software that needed to be produced quickly. Willem suggested that he use this as an opportunity to create a high-performance team, practicing the latest software development methods. No one would be assigned to the team. Instead, they would opt in themselves.

As Willem expected, developers, testers, analysts, and others were excited to opt in to this new kind of team. Their managers reluctantly agreed, even though they were concerned that this initiative would get in the way of their management responsibilities (that is, to be *in control*) and thus hurt their standing in the organization.

Willem worked with the opt-in individuals for months, helping to mold them into a high-performance team. Together, they produced a high-quality, easy-to-maintain software solution in record time.

Once the new software was in use, the team was disbanded. Each team member was asked to come back to their original team and their original manager. And once they returned, none of the cutting-edge planning, teamwork, or software development skills they had learned were used.

Willem's successful experiment was completely ignored by the company's management! He was astonished. Wouldn't these managers want to repeat the success his team had just delivered? Or even leverage parts of it?

Apparently not. Managers didn't feel like the success was theirs because they had little to do with it. Meanwhile, they were focused on the initiatives for which they could be rewarded or recognized by upper management. They were relieved to get their people back to work on those initiatives.

Change is tricky. When people aren't motivated to change, even successful experiments will fail to make a difference.

Motivation to change is essential to making real improvements.



Changing Our Perspective on Change

Most of us resist change. Why is that?

Many years ago, I learned one answer to that question by participating in an exercise called Change Five Things. Since then, it has become one of my favorite exercises because it's so successful at helping people understand a primary reason why most of us resist change.

The exercise begins by asking participants to find a partner. Next, they are requested to face each other and observe their partner's clothing, jewelry, and whatever else they are wearing. After a minute or so, they are told to turn away from each other and change five things about their own appearance. When that's done, they turn back around to face each other once again and each participant gets a chance to identify what changes the other person made.

Immediately after participants have completed that part of the exercise, they are asked to turn around once again and make five *additional* changes to their appearance, without undoing any changes they made in the first round. When five new changes have been made, they turn around and again try to identify them.

Finally, they are asked to do it one more time! This is when a few participants tend to shout out, "I don't have any more clothes to remove!" or "This could become an HR incident!"

These objections are an excellent cue to stop the exercise and begin the all-important debrief. During the debrief, participants recount that during the exercise, they took off sweaters, belts, socks and shoes, and removed precious jewelry such as wedding bands, necklaces, and earrings.

I ask everyone, "What if we continued to run more rounds of this exercise?" Many people say, "We'd be naked!" This is the perfect time to reveal what the exercise teaches, but first I describe my own experience.

"When I participated in this exercise," I tell the participants, "I did exactly what you all did: I removed lots of items." Then I ask, "So why is it that when I asked you to change five things, you *removed* things? Why, for example, didn't you find things in the room and *add* them to your appearance?"

I demonstrate that there are plenty of items in the room to add to one's appearance, such as pieces of paper, pens, computer cables and mouse pads, and so on. I ask, "Why do we naturally tend to *remove* items during this exercise instead of *adding* them?"

And then comes the answer:

Because humans associate change with loss.

Wham.

That really hits people because they just physically experienced the loss of their own items when asked to change. It's a profound lesson that's hard to forget. It helps us understand how our colleagues, friends, or family resist change, as well as an opportunity we have to begin to associate change with gain instead of loss.

I've conducted Change Five Things with thousands of people around the globe. Once, I ran an executive team through the exercise before beginning their company's agile transformation. A week later, one of the executives contacted me and explained how grateful he was because the exercise helped him deal with a painful personal situation.

Change does not have to equal loss. If you can change your perspective about change, you may discover how change can produce gain.



Get Comfortable with Being Uncomfortable

In January 1996, I left a comfortable, six-figure job on Wall Street to start my own company. I had no background in business, no clients, and no clear idea of what my new company, Industrial Logic, would do.

Fortunately, I'd spent the previous six months talking with a therapist about this major change I was planning to make. Through these discussions, I learned that I would need to become "comfortable with being uncomfortable" in order to be successful.

The first year is brutal for most new businesses, and my business was no exception. I had no work, I had no income, and I didn't yet know where my customers were going to come from. To help turn my new business into a success, I deleted *Doom* and *Asteroids* from my computer, and I read a lot of books—especially those on how to run a business.

A couple of months after I started up Industrial Logic, I met someone from MTV at a party and I gave him my business card. A couple of months after that, he emailed to see if I wanted to bid on some work. The work turned out to be building a Java version of the MTV website. I jumped at the chance.

I spent the next three months trying to win the work via countless meetings and design presentations at MTV's offices in Manhattan. I later learned I was competing against two large, well-established companies. To my surprise and delight, I won the bid. Everything got a bit easier after that.

I'll never forget that first year navigating through the deep fog of uncertainty and leaning on what I'd learned from my therapist. I made less than \$20,000 that year, but I was thrilled. I was becoming comfortable with being uncomfortable, and my business was growing.

Big changes aren't comfortable. It helps to get comfortable with being uncomfortable.



Learned Helplessness

Tom, an extremely intelligent, mathematically gifted engineer and software developer, was unhappy. He and his team had moved from individual cubicles on the second floor of their office building to an open workspace on the first floor, and Tom didn't like it one bit. I knew this because Tom had mentioned his displeasure with the move to the open workspace during his team's end-of-iteration retrospective.

I was engaged as an external coach to help Tom's team and several others. We conducted retrospectives at the end of each week. Each team would gather around a board, with two columns labeled "Working Well" and "Needs Improvement." Members of the team would add items to each column, we would discuss the items, and then work on ways to solve any issues that needed improvement.

During his team's first iteration retrospective, Tom added "Open Workspace" as an item under "Needs Improvement." When the team discussed Tom's issue, someone asked him for more details. "What specifically isn't working for you, Tom? Can you give us a few more details?"

All Tom could say was, "I don't know. I just don't like it."

By the end of the team's third iteration retrospective, Tom's item remained unsolved. I personally asked Tom if he could provide any more details about what he didn't like about the open workspace or suggestions for how to improve it.

All he could say was, "I just don't like it."

That bothered me. An agile team is meant to learn and adapt rapidly, yet here we were, stuck on an issue. I decided to try a one-on-one discussion with Tom, just before lunch, when few people were around. I asked him if he could provide any more details on what wasn't working for him about the open workspace. Finally, he opened up!

Tom showed me his awkward seating arrangement. His workstation was positioned near a cubicle wall. "Whenever people walk behind me," he said, "I have to scoot my chair in, so they can get by and then my knees hit this column."

I said, "Oh, that's awful. I can see how uncomfortable that is!"

Then Tom explained that in his old cubicle on the second floor, he had plants and pictures of his family, whereas in the open workspace, he had none of those. I acknowledged those problems, too.

Then I asked Tom, "Where would you like to sit in this open workspace?"

Tom immediately pointed to a window and said, "Over there, by that

window.”

I thanked Tom for speaking with me, and he went off to lunch.

Tom was not being resourceful. Worse, he was being helpless. He was perfectly free to move his desk and bring his plants and family photos to the open workspace. But he hadn't felt empowered to do it. Why?

Learned helplessness, a term coined by American psychologist Martin Seligman, is a common problem. People who suffer from it become powerless to make changes, often based on prior negative experiences.

Tom was an outstanding performer on his team and was well regarded within his company. It puzzled me why such a talented person was helpless to improve his seating position and physical environment.

I wish I had helped Tom explore and understand what was holding him back. Had he stuck his neck out in the past and been punished for it? Had he experienced a negative interaction about the new open workspace before the move? Was he afraid of someone in his own team? I didn't know, and with a limited number of coaching days for helping several teams, all of which had problems to address, I didn't believe I had time to find out.

What I did know was that it's bad practice to leave a problem unresolved for many weeks on a team's retrospective board. Tom's problem with the open workspace needed to be fixed and I wanted to demonstrate being resourceful to Tom and the team.

There were many empty desks on the first floor, so it was not hard to move one next to the window that Tom liked. The trouble was that we would need a long ethernet cable to connect Tom's computer to a router in the center of the room. I found a facilities person and asked how long it would take to obtain a twenty-foot ethernet cable and get Tom connected.

“Three weeks,” was the response.

“Never mind,” I replied.

I immediately jumped in my car and bought a twenty-foot ethernet cable at a local electronics store. Driving back to the office, I saw someone selling plants on the side of the road. I pulled over and bought a few plants. When I got back to the team's workspace, I asked someone to help me move one of the spare desks to the window location that Tom liked. Then I moved Tom's computer on top of the desk and connected it to the router via the ethernet cable I had bought. Finally, I put the newly purchased plants onto Tom's desk.

When Tom returned from lunch, he was utterly amazed and delighted. At the next iteration retrospective, Tom's card about the open workspace moved from “Needs Improvement” to “Working Well.”

I was both happy and sad. It was great that Tom was now more comfortable

in his workspace, but I was the one who was resourceful, not Tom. Why was he so powerless to fix his own problems? I wasn't resourceful in helping to get to the root of that problem. It was a missed opportunity to help Tom (and perhaps others) become resourceful.

Learned helplessness is poison to resourcefulness. Invest time in understanding and overcoming it.



Be Resourceful, Not Resentful

In the early 1980s, when I was a teenager, my family lived in a tennis-obsessed community in Long Island, New York. Public tennis courts were everywhere and were usually packed with players. At the time, tennis was extremely popular around the world. Millions of fans watched Bjorn Borg, John McEnroe, Ivan Lendl, Chris Evert, and Martina Navratilova engage in epic tennis battles to win championships and become number-one ranked in the world.

The Port Washington Tennis Academy, where McEnroe had once trained, was only a few miles away from our house. Competition to make my high school tennis team was fierce. Only athletic kids who took tennis lessons for years made the team. I wanted to be one of those kids, but since my parents couldn't afford tennis lessons, I resigned myself to playing tennis recreationally and watching it on TV. Instead of being resourceful, I was mostly resentful that my family couldn't afford tennis lessons.

One spring afternoon, toward the end of my senior year in high school, some friends of my parents came over to our house. Sitting in the backyard, the subject of tennis came up, and I mentioned to these friends that, while I loved tennis, we couldn't afford tennis lessons, so I never competed at the high school level. Ingeborg, one of the friends, then asked me a question that I will never forget: "Why didn't you pay for the lessons yourself?"

Her question stunned me. I had no good answer! I regularly made money by mowing lawns, raking leaves, shoveling snow from people's driveways and walkways, babysitting, and even writing software. Why had I not paid for the lessons myself? If I found a tennis instructor and explained that I would be paying for the lessons myself, the instructor might have offered me a discount or given me jobs to perform at the tennis academy to lower the cost of the lessons. I could have even ridden my bike to the Port Washington Tennis Academy!

Ingeborg's question changed me. It was a light-bulb moment. A few years later, when I'd finished college and was working in NYC, I paid for my own tennis lessons.

Being resentful is a clue that it's time to become resourceful.



Got a Coach?

It's not how good you are now, it's how good you're going to be that really matters," said surgeon and Harvard Medical School professor Dr. Atul Gawande in a 2017 TED talk watched by millions. But how do we know when we are no longer getting better at our jobs? Frequently, we don't, so we need some way to go beyond the state of no-improvement.

In the conventional view, professionals are responsible for their own improvement—going to school, getting some training, and making their way in the world. It's up to you to manage your own improvement, and many, if not most, professions use this approach. Gawande observed that the field of sports is a notable exception—it's very common practice for athletes to have coaches. To take just one example, professional football teams often have a head coach, assistant coaches, a defensive line coach, an offensive line coach, a quarterback coach, a special teams coach, and many others.

Gawande decided to ask the virtuoso violinist Itzhak Perlman if coaching could work outside of sports, in this case, in music. In a nearly two-hour phone conversation, Gawande learned that Perlman had a coach—his wife, Toby! She was also a concert violinist and had graduated from Julliard, but gave up her own career to be her husband's coach. While attending his concerts, she would listen carefully and give him constructive feedback afterward. Perlman explained that his wife's coaching was crucial to his continued development and success.

Gawande said when we try to do everything ourselves, there can be a number of problems that might block our growth. One is not understanding the issues that are in our way or not knowing how to resolve them. The result is that we stop improving.

In a stirring moment of transparency, Gawande revealed that he noticed he had stopped improving in his own surgical practice. His practice began in 2003, and for the first several years, he was improving steadily—his complication rates were dropping year after year. After five years, however, Gawande's complication rates plateaued, and a few years later, he realized he was no longer improving. That's when he decided to try a coach.

He asked a former professor of his, Dr. Bob Osteen—a general surgeon who lives in Boston—to observe his work and critique it to see if it would help him improve. Osteen agreed. He observed Gawande in the operating room with a patient, a procedure Gawande thought went beautifully.

Not exactly.

Osteen had filled a page with notes about "small things" he had observed. A

light near the wound moved out of position, one of Gawande's elbows periodically went up in the air, which indicated he didn't have complete control. Gawande wasn't aware of these minor occurrences when they were happening. Great coaches, he explained, provide a more accurate view of reality to help improve performance.

The experience of having Osteen in the operating room and giving him feedback was a profound one for Gawande. He learned he needed coaching to continue getting better as a surgeon. In just two months of coaching, Gawande noticed he began improving, and after a year, his complications decreased.

The experience of being observed was a painful one, said Gawande, and he didn't want to work on some of the things he needed to, but he had seen the power of coaching for himself. It was undeniable.

Who's helping you learn and improve? Sometimes the best way to be resourceful is to get a coach.

● ● ● Engage an Ensemble

When Thomas Jefferson, the third president of the United States, wrote the first draft of the Declaration of Independence, delegates to the Second Continental Congress debated the language in his draft and suggested changes. Benjamin Franklin, himself a delegate, suggested changing the words “We hold these truths to be sacred and undeniable” to “We hold these truths to be self-evident.”

Jefferson, an excellent writer, felt uncomfortable with the many suggested changes. Franklin sought to console his good friend by telling him the tale of John Thompson.¹

John Thompson had just started out in the hat-making business and wanted a sign for his shop. He composed his sign like so:



JOHN THOMPSON, HATTER, MAKES AND SELLS HATS—FOR READY MONEY

Before using the new sign, John decided to show it to some friends to seek their feedback. The first friend thought that the word *Hatter* was repetitive and

unnecessary because it was followed by the words *makes hats*. The word was struck out. The next friend observed that the word *makes* could be removed because his customers would not care who made the hats. So *makes* was struck out. A third friend said he thought the words *for ready money* were useless, as it was not the custom to sell hats on credit. People were expected to purchase hats with money. So those words were omitted.

The sign now read, "John Thompson sells hats."

"Sells hats?" asked his next friend. "Why, nobody will expect you to give them away. What then is the use of that word?" *Sells* was stricken. At this point there was no use for the word *hats* since there was already one painted on the sign. The sign was ultimately reduced to:



JOHN THOMPSON

The hatter's sign was now simple and elegant. By engaging a small ensemble of friends, John Thompson ended up with a far better sign for his store.

The summer before my freshman year of college, I was required to read Homer's *Iliad*. On backpacking trips that summer, I often read the book by flashlight. After finishing the book, I considered it to be just a famous old war

novel.

When I got to Saint John's College in Santa Fe, New Mexico, and attended my first seminar on *Iliad*, I realized that the book was far more than a war novel. In the seminar, our group of eighteen students and two professors explored ideas like fate versus free will, examining passages in the book and peeling back layers of meaning that I hadn't realized were there. It was exhilarating and an unforgettable lesson in the power of a group deriving far more understanding of something complex than any lone individual.

Early into my career in the software field, I applied what I'd learned at Saint John's College to host study groups on important literature in software design. Small ensembles of programmers would meet weekly to discuss and learn about influential books and papers. It was an effective way to learn deeply about subjects that were critical to our profession.

Hit comedy shows start life in rooms filled with writers who banter, crack jokes, suggest ideas, shoot ideas down, and build on new ideas. Together, they collaborate to make great content. Being funny show after show is hard. No single mind can do it alone.

In software development, ensembles of programmers, testers, analysts, product managers, and subject matter experts, working closely together, often write better quality code faster than individuals working solo. That's because the hard work is in thinking and designing, not typing the code. Engaging an ensemble in programming enables faster feedback and a continuous integration of ideas.

Whether you are endeavoring to understand complex ideas or create something important, engaging an ensemble can yield better results than individual effort.

Produce higher-quality results by engaging an ensemble.

1. tjrs.monticello.org/letter/27.



How Pete Sampras Won Big by Learning to Lose

Whispers in the fiercely competitive junior tennis community began soon after Pete Sampras switched to a one-handed backhand. Suddenly, the hardworking teenage tennis sensation was losing! Not only was he losing, but he was also losing to opponents he once routinely beat.

Why had Sampras made such a radical change? His strong, accurate, and reliable two-handed backhand had been working well, and now his junior rivals were hammering his inaccurate and unreliable one-handed backhand. Andre Agassi, another rising tennis star, said at the time that he felt bad for Sampras.

The change to the one-handed backhand was championed by an unlikely tennis coach named Pete Fischer. A successful endocrinologist and pediatric growth specialist, Fischer was a heavyset man with a big belly who belonged to the same tennis club where Sampras played as a teenager. Sampras said that Fischer had “one of the most horrific tennis games anyone ever laid eyes on.”

What he lacked in tennis skills on the court, however, Fischer made up for in his knowledge of tennis strategy and his visionary approach to developing Sampras into a champion. In his memoir, Sampras said of Fischer:

In retrospect, “coach” is not exactly the right word for Fischer, because his greatest asset was knowing what he didn’t know. He was a hacker tennis player who masterminded my tennis development in a wise way—by having various coaches and specialists bring their unique skills to my development. He had grand, almost preposterous plans for me. He was like a combination of mad scientist and general contractor—one who was in charge of building the all-time Grand Slam champion.¹

Fischer helped Sampras develop his serve, and like a general contractor, selected legendary coaches to train Sampras on groundstrokes, footwork, balance, and volleys. This group of coaches was instrumental in developing Sampras into a champion. Yet, in his memoir, Sampras said that the pivotal moment in his development happened when he was fourteen and Fischer persuaded him to switch to a one-handed backhand.

Fischer had a vision for Sampras to become the next Rod Laver. That meant winning many Grand Slam tournaments, including Wimbledon. At that point in tennis history, the thinking was that a champion needed a one-handed backhand to win on Wimbledon’s fast grass courts.

Sampras trusted Fischer and agreed to make the risky switch. It was an

exceedingly hard change to make, especially for a teenager involved in fierce competition. Sampras said, “No junior player in his right mind, and none that I know of, was willing to take a risk of that magnitude.” The change led Sampras to lose to all kinds of players. His winning tradition evaporated, and people were talking. Most painful of all was losing to his archrival, Michael Chang.

What enabled Sampras to endure all of this losing? His upbringing certainly helped. Sampras said,

*If I got my calm, reserved nature from Dad, I got my toughness, a share of my resilience, and a measure of my stubbornness from my mom. She helped instill my basic values—showing me that I wasn’t going to get anywhere by taking shortcuts.*²

Fischer also helped enormously. Most tennis players, especially juniors, have fragile egos and hate the humiliation of losing. Fischer helped Sampras understand that losing to his junior peers while he developed his one-handed backhand didn’t matter—they were shaping his game for championship play at the professional level. Sampras said, “By putting pressure on myself to develop a great game, I had less pressure to win.”

Learning to lose in his junior years without having his spirit or confidence broken would help Sampras throughout his entire career. Sampras explained how the strategy that he and Fischer developed was the opposite of most juniors:

*To us, it was always about playing the right way, trying to develop a game that would hold up throughout my career. It was a calculated risk. If it didn’t pay off, it would have shown that I was either not good enough or delusional. On the other hand, some of those juniors were like starving guys, eating everything on the table while the eating was good. They didn’t think long term, they lived and died by their daily results, ignoring the fact that what worked in the juniors (like endless soft, arcing rally shots) wouldn’t necessarily be useful on the pro tour.*³

In his book *Unlearn: Let Go of Past Success to Achieve Extraordinary Results*, Barry O’Reilly explains that finding higher levels of success often requires unlearning the habits and behaviors that worked for us in the past. It took Sampras two long, painful years to unlearn his reliable two-handed backhand and turn his new one-handed backhand into a reliable and formidable weapon. Once he did that, Sampras was unstoppable. He reached a whole new level of tennis greatness and went on to win many Grand Slam tournaments, including seven men’s singles titles at Wimbledon. In the 1990s, he held the number-one ranking in tennis for six straight years, a record that has yet to be broken.

Focus on the long game. Learning to lose can help you win big.

[1.](#) Pete Sampras and Peter Bodo, *A Champion's Mind: Lessons from a Life in Tennis* (New York: Crown, 1st edition, 2009).

[2.](#) Ibid.

[3.](#) Ibid.

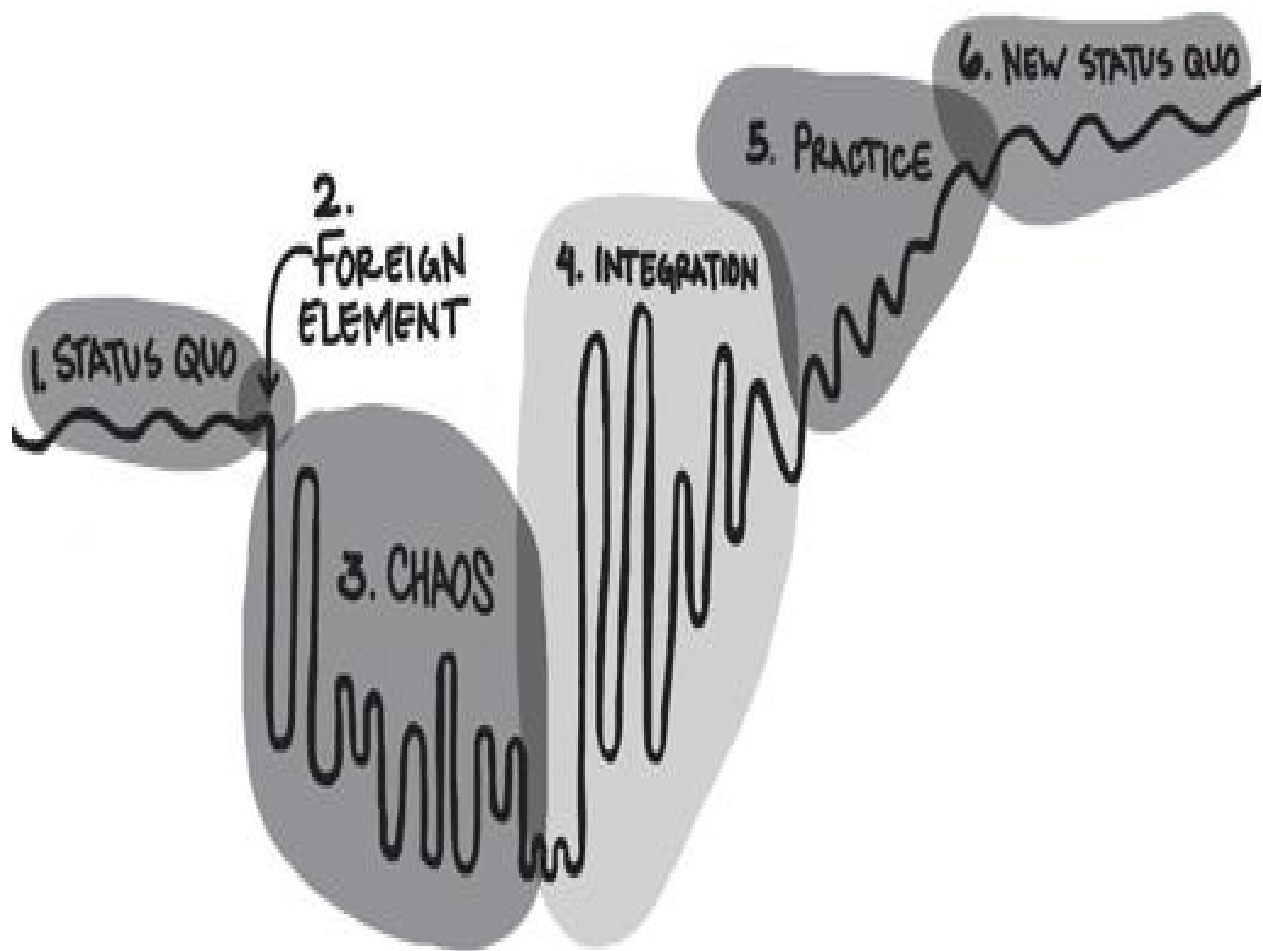


The Change Curve: Virginia Satir's Six Stages of Change

In her famous model of change, Virginia Satir—an innovative family therapist, author, and teacher—described the stages we go through whenever we experience a significant change in our lives. While her context was family therapy, her insights about change are applicable to anyone experiencing or facilitating change.

In their book *The Satir Model*, Satir and her coauthors acknowledged Newton's First Law of Motion: *Matter continues in its state of rest or uniform motion unless it is compelled to change by forces acting on it*. Since most of us are comfortable with our familiar behavior or normal performance, change often requires strong motivators to move from the familiar to a higher level of functioning. Such motivators may be external (for example, a job loss or recession) or internal (for example, pain, fear, or hope).

Satir's change model consists of six stages that describe what happens to people who *want* to change. Since performance tends to worsen before it becomes significantly better, it has become popular to illustrate the six stages as a curve:



Here's a description of each of the six stages:

1. **Status quo** is when “the need for change first emerges.” A person or system is operating within a normal, stable, and predictable pattern, when something happens to trigger a desire to change. The trigger could be a recognition of an unmet need, a painful event, a newfound threat, or awareness of a potential benefit. While the name “status quo” implies that nothing much is changing, this is in fact when change first begins.

2. **Introduction of a foreign element** involves “the system or individual articulating the need for change to another person—friend, therapist, or someone else outside the system.” At this point, resistance and self-protective behaviors may be strong, especially if the change appears to remove protections or threaten survival. During the early part of this stage, the person or system has a “natural tendency to continue its status quo and may in fact want to remove the foreign element.” Satir understood that *reframing* was essential at this point. She would work with clients to “see themselves as capable and open to change” rather than resistant to it. She coached them to acknowledge old habits as friends, thank them for what

they had done, and then let them go.

3. **Chaos** is the next stage and it occurs after the successful entry of a foreign element. It involves “moving from a status quo into a state of disequilibrium.” Now the familiar and predictable become unfamiliar and unpredictable. During this stage, levels of anxiety may reach their peak. Fear of the unknown may lead people to feel a loss of safety and stability or question their self-worth. A change agent (therapist, coach, etc.) must be cool, calm, and collected at this time, help “neutralize clients’ fear and anxiety,” acknowledge their stress, and encourage them to open up to the possibilities of something new. Chaos is when growth begins.

4. **Integration** is next. It’s a period during which “new learnings are integrated, and a new state of being evolves.” We “take charge of what we once did on automatic pilot, bringing our control back into consciousness, and becoming more responsible for our internal process of Self.” During this time, people may feel moments of joy or excitement as they catch a fleeting glimpse of a better way of being or performing. They are beginning to experience what the new status quo may be like.

5. **Practice** is the next stage, a time during which the “new state is strengthened by practicing the new learnings.” It’s an essential time for transforming what was once strange into something familiar and comfortable. Falling back to older ways is still possible. Satir and coauthors cautioned, “We need to remember that the power of past patterns is strong.” Heightened awareness is critical at this stage. Satir recommended using reminders, exercises, visual cues, or whatever could help remind people to stay focused on practicing the new way.

6. **The new status quo** is the sixth and final stage. The person or system now exhibits a more functional state of being. This new way is predictable and reliable. It provides a sense of harmony and wholeness. Attaining this level opens people or systems up to further growth rather than getting stuck with the familiar. It’s a stage of enhanced well-being. And it’s not the end of the road. A new opportunity for improvement may occur when the new status quo becomes the old status quo.

Satir’s change model has helped me understand, accept, and navigate change in my personal and professional life. It has made it acceptable to dwell within a period of chaos before attaining a higher level of functioning. It’s an invaluable guide for anyone who’s interested in change.

Let Satir’s change model be your guide in successfully navigating change.



The Change Curve for My Tennis Serve

In my early twenties, I personally experienced Satir's change curve when I made a total and complete change to my tennis game.

At that point in my life, I'd never received formal instruction in the sport, and my game was mediocre at best. One day, while playing on a public court in New York City's Central Park, I lost a match, largely due to my terribly inconsistent serve. As I walked off the court, an instructor approached me and asked if I'd be interested in a few free tips to improve my serve. I was happy to take the advice, so the instructor and I walked back onto the court.

The instructor then asked me to hit a few serves so he could observe my service motion. Back then, I used to try to emulate John McEnroe's service motion. I'd stand with my feet sideways against the service line, slowly lean forward and down, rise back up, then launch the ball in the air and drive my racquet against the ball as my body fell forward. I rarely got the ball in on the first serve and had to hit a lame "dink" shot to get the ball in on the second serve. In other words, my serve sucked!

After watching my amateur foolishness, the instructor said, "My first advice to you is to *stop moving so much!* You will *never* have a consistent serve if you move your body so much!" He suggested simply standing upright, then launching the ball into the air and hitting it.

I tried his suggestion and immediately found that it was easier and reduced errors. Lesson learned.

Next, he looked at how I was holding the ball when I tossed it into the air. He suggested placing my thumb on the north pole of the ball and my four fingers at the south pole, before launching the ball into the air.

I tried this technique and immediately found that it wasn't hard to do and seemed to improve my toss. Another lesson learned.

I was quite pleased with the instructor's suggestions thus far, but that was about to change. The instructor next asked me to show him how I gripped the tennis racket. I had learned years earlier to grip the handle of the racket the same way I would shake someone's hand. The instructor explained how this was all wrong—none of the pros held their rackets that way.

He suggested I turn my wrist ninety degrees counterclockwise and then showed me how to pronate (a new word for me back then) during the service motion. He demonstrated the technique and his serves were indeed marvelous.

However, when I used the new grip and tried to pronate, it was a disaster! I hit balls far into other courts and had to wave at distant tennis players to return my ball. I tried several times to use this new grip and to pronate during the service motion. It felt totally alien. I was serving far worse with this approach than before I'd met the instructor!

The instructor explained that I would need a lot of practice with the new grip before it would seem natural. He suggested lessons. I had seen how well he served and wanted to serve like he did. I bought a package of ten lessons.

During my first lesson, I continued to feel incredibly awkward holding the racket with the new grip. Balls flew into other people's courts. Perhaps once or twice during the hour-long lesson, I hit a serve that wasn't awful. But, overall, I wasn't happy with the trend.

During the second and third lessons, there were small improvements. More balls stayed in the court and my service grip was no longer feeling utterly alien. By the fourth and fifth lessons, I began to see real improvements. I was learning how to actually pronate, and some of my serves felt smooth and easy. The ball would not only stay in the court, but it would also land in the correct box on the other side of the net.

I practiced a lot, and by the sixth and seventh lessons, I was experiencing a totally different serve. It was far better and more consistent than my old serve! The instructor began teaching me how to hit a proper second serve as well.

Lessons eight, nine, and ten helped me solidify what I'd learned and work out some kinks. When I completed all ten lessons, I was delighted with my progress.

Now, when I played matches, my serve was in an entirely different league. It was a dependable weapon. I still made errors, but I knew how to diagnose them and make adjustments. My new serve was vastly superior to my brother's and the friends' I'd grown up with playing tennis.

The small changes suggested by the instructor, such as limiting movement during the service motion and gripping the ball differently during my toss, had improved my serve a little. But breaking through to a much higher level of skill required me to experience the six stages of change described by Satir. And that experience transformed my life well beyond tennis.

Years later, as I worked with clients to significantly improve their software development skills, I'd share my tennis serve story and help them appreciate and accept the need to become worse before becoming far better. Not every client was ready for that. Some were content with a series of smaller changes. But others wanted significant change and were happy to learn new skills that required the journey from worse to far better.

If you want to become significantly better at something, be willing to become worse before you become far better.



Celebrate Chaos

The CTO of Kronos, Peter George, knew that the five-hundred-person team that built and maintained the company's time and payroll management software needed to improve. The suite was used daily by more than thirty million people in companies and governments around the globe. Version 5.1 was supposed to have been delivered to customers in eleven months, but instead took fifteen! To make matters worse, only seven of those fifteen months actually involved new development, while a whopping eight months was spent on "hardening"—a process of removing new defects—struggling to integrate code written by many different teams, and manually testing that everything worked.

Peter was tired of the delays, quality problems, and stress. He called for outside agile consultants to help.

Jim Highsmith, an agile expert, and I met with Peter and his team to understand the company's challenges and aspirations. Over the ensuing weeks, we conducted an assessment and formulated a plan to help Peter's team shift from its traditional process to an agile one. Our plan involved significantly reducing hardening time by introducing agile practices to accelerate collaboration, automate testing, reduce defects, and make integration continuous. The changes we proposed would impact development, management, planning, even performance reviews. We suggested a gradual rollout of the agile process—starting with a few pilot teams, expanding to more teams, and then eventually extending to all teams.

George liked our plan, except for the gradual rollout part. Instead, he wanted all five hundred people (including product management, architecture, development, testing, interaction design, and technical writing) to become agile as quickly as possible so the company could focus on high-value work instead of wasting so much time in hardening activities.

Jim and I cited numerous risks with George's big-bang idea. George countered by explaining that Kronos's software consisted of twenty tightly integrated products. Since his teams were all deeply intertwined and dependent on one another, if only a subset of these teams initially learned the agile way of working, their interactions with the other teams would be slow and difficult. He had a point, so we agreed to work with the entire team from the start, even though this was unusual for us and took us far out of our comfort zone.

To make the plan work, Jim and I designed something we called "Agile Lite," a subset of essential agile management, planning, and programming practices. Together with our team of agile coaches, we would help Peter's team

reach a higher level of functioning by learning and practicing Agile Lite during the first wave of a multi-wave agile transformation. If Agile Lite succeeded, we would devote later waves to helping teams mature in their agile practice. George liked the idea, and we agreed to get started.

The work began with an employee all-hands meeting in the cafeteria. George explained how the 7:8 ratio of new development to hardening time was hurting the company's ability to satisfy customers and compete in the marketplace. He set a bold new challenge of getting to an 11:1 ratio within a few years.

After Jim Highsmith explained Agile Lite to the large group of employees, it was my turn. I began by describing my worse-to-far-better journey to significantly improving my tennis serve and how that experience mapped to Virginia Satir's change model. I showed a graphic of the Satir curve and explained how our agile journey would take us from the status quo into a period of "chaos" where we would feel and be less productive. I explained that this period of chaos would pass as we integrated and practiced new ways of working that could ultimately help us achieve a consistently higher level of performance.

The weeks that followed were intense. Since there was neither time nor budget to train everyone, a small subset of people received basic training in agile management, planning, and programming. New, balanced teams were formed and the agile coaches from my company helped people on many teams. For example, product managers and analysts learned to split large features into smaller, incremental versions and regularly update release plans. Technical staff learned how to integrate software several times a day and apply test-driven development, a practice that helps produce code with few, or even no, defects. Each week, we sent out surveys to the entire five-hundred-person team to understand what was working well and what needed improvement.

Not everyone liked the changes. Employees would regularly stop by George's office to complain. Some questioned the continuous integration approach. They speculated that after a month of work, it could take the five-hundred-person team two to three weeks of effort to produce a successful build of the entire suite, squandering precious development time. George simply asked them to give it a try.

Two months into the transformation, when everyone showed up for work at the two-story office building on the outskirts of Boston, there were balloons everywhere.



What were the balloons for? No one knew.

A few hours passed. Finally, Peter sent out an email to explain the balloons. “Congratulations Everyone!” began the email. “We are officially celebrating being in chaos! This is exactly where we wanted to be, since we knew we needed to get worse before we could get significantly better. We look forward to the next steps in our journey!”

It was a marvelous communication by a CTO who really understood how to lead people through change. I asked Peter to describe the signs of chaos he noticed. He explained that people were coming to his office and complaining that the coaches were challenging them to integrate work far more frequently than they ever had before. In addition, employees complained that the coaches were asking them to test-drive code and produce dependable suites of automated tests, and they were also expecting them to collaborate in ways that were entirely new—making the employees feel uncomfortable. George then said, “These kinds of complaints are far better than the normal complaints I get, so I am extremely pleased. Please carry on with your work.”

Celebrating chaos was a clear message to everyone that Kronos was committed to improving. As the months passed, the teams integrated and practiced what they had learned from our coaching. They improved at planning, testing, developing, and building new software releases. Within six months, they could produce weekly builds of the entire time management suite and had automated much of their complex build process. A large pool of older defects and extremely complicated code continued to slow them down, but progress was being made.

When the new release was done, it had taken twelve months—resulting in an 8:4 ratio of new development to hardening time. While they had not yet reached George’s vision of an 11:1 ratio, they were making excellent progress. The first wave of a multi-wave agile transformation was done. Kronos was operating at a new status quo and ready to begin making more improvements.

Celebrate chaos on the journey to better performance.



Make Problems Visible

“Don’t Tolerate Software Crashes!” was the headline of a full-page advertisement in the leading magazine for mass spectrometers—a machine commonly used for chemical analysis. The words were in a very large font and the ad featured a woman holding her hand to her forehead, symbolizing the headache this software was causing her. This ad was a direct attack on the industry’s leading maker of mass spectrometers, a manufacturer that had just hired my company for our agile software services.

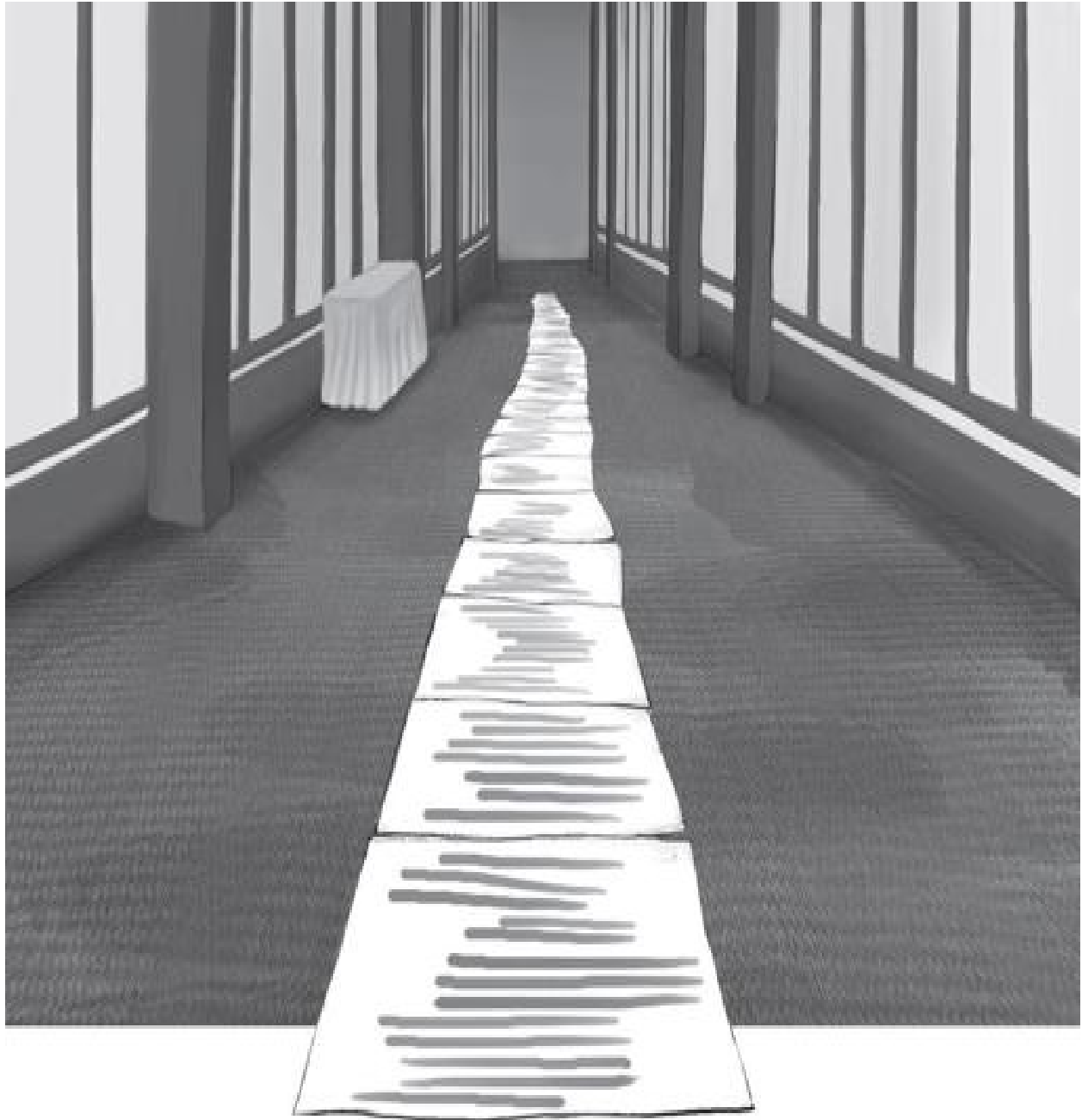
The code causing these crashes was a mess. It had all the hallmarks of bad software design. The programmers who maintained this code generally looked miserable, spending hours each day grappling with extreme complexity. They were consumed by fear since any change to this fragile codebase could easily produce new defects.

Surprisingly, management didn’t seem to understand the magnitude of the problem. They wondered why the programmers were so slow. And to make matters worse, the product manager had a long list of new features to add to this product.

New features? When the system wasn’t even performing the basic function for which it was sold? When customers were regularly experiencing debilitating software crashes?

No, the new features could wait. It was time to focus exclusively on fixing the quality and stability problems.

Quality code tends to have small functions, each of which performs one responsibility, thereby making it easy to produce automated tests that check whether the functions work correctly. Large functions, which contain many computer instructions, are notoriously hard to test and prone to defects. I asked the programmers for a printout of the longest function in the codebase.



That evening, over drinks with a colleague in our hotel lobby, I decided to tape all twenty-eight printed pages of code together to form a giant scroll. Then I rolled it out on the lobby floor, grabbed my phone, and made a short video about it.

In a meeting with the company's leaders the next morning, I showed them the video. Finally, they grasped the true weight of the problem.

My colleagues and I immediately turned our attention to helping this company fix the problem. We started with a clear objective: the mass spectrometer must be able to acquire data for forty-eight continuous hours

without crashing. Over the next six months, we helped the team collaborate closely to automate tests, fix defects, and simplify code.

When the new version of the product shipped, it was orders of magnitude more stable. Thank-you notes from customers poured in. Stability was the feature customers wanted most and competitors wanted least.

So why didn't managers focus on this problem sooner? It was invisible to them, an unarticulated customer need. That's why it's vital to raise awareness of problems, like erecting warning placards around a pool of spilled liquid in a cafeteria:

So, whether you've got a bottleneck in your workflow, too much work in process, slow handoffs between teams, too many interruptions, poor quality, or something else, it can help to make the problem(s) visible. In her excellent book *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*, Dominica DeGrandis—a lean/agile expert—explains not only how to make problems visible but how and why to make work itself visible. In a foreword to Dominica's book, Tonianne DeMaria Barry, herself a lean guru and coauthor of *Personal Kanban*, made this observation about making work visible:



It should come as no surprise that we can better manage what we can see. When we can't see our work, our options are obscured. We're blind to our own capacity and we certainly can't communicate that capacity to others. The resultant mental overload creates stress. Stress compounds the work we already have, essentially contributing to WIP, compromising the ability to focus, prioritize, make decisions, and complete work with quality, let alone complete work at all.¹

Make problems visible so others can see the need for change and take action.

[1.](#) Dominica DeGrandis, *Making Work Visible: Exposing Time Theft to Optimize Work & Flow* (Portland, OR: IT Revolution Press, 2017).

● ● ● Halloween in April

When one of my daughters was four years old, she was invited to the birthday party of a friend who was turning four. The birthday boy's parents knew that their son loved Halloween but his birthday was in April, not October. Showstopper, right?

Not for these parents!

They simply decided to *fake* Halloween, lock, stock, and barrel! They bought lots of candy, distributed it to several neighbors who kindly agreed to go along with the fun, and then six four-yearold kids went house to house, trick-or-treating in April!



In all of my years of parenting, whether hosting birthday parties for my three kids or taking my kids to birthday parties, nothing compares to the creative resourcefulness of these parents. Instead of accepting the fact that Halloween only happens once per year, they surprised and delighted their son by creating a

fake Halloween. It was awesome.

If you've got a constraint, problem, or challenge, be creative in overcoming it.



Tear Down the Cubicles

Standing in front of a room of a few dozen programmers, I asked, “How many of you are actively using what you learned in the workshop I gave six months ago?” Only one hand went up! I was stunned.

Six months earlier, these programmers had taken my best-selling workshop and given it extremely high ratings. The host organization was so pleased with the training that they invited me to come back and teach it again. I agreed, provided that after the second workshop I could interview the first batch of students and get a chance to talk with management. They accepted those terms.

The second workshop went as well as the first one did, and now I was interviewing the first batch of students. The one programmer who’d raised his hand explained that the reason he was using what he had learned in my workshop was simple: he didn’t have a manager, so he was free to work as he saw fit. Meanwhile, he explained, the other programmers who attended my workshop had managers who refused to let them change how they developed software. They were not allowed to use the critical software engineering skills I’d taught them.

This was bad. The company was struggling with deeply complex, highly fragile, and buggy software. It cost them as much to maintain the code as it did to build it. Their customers were unhappy with the software and they were getting increasingly vocal about it. The company desperately needed a better way of working, and middle management was preventing the developers from trying a new way.

I concluded the meeting with the first batch of developers and then welcomed management into the room. I was not happy. Nearly fifty programmers from this company had just taken my workshop, and yet the business would not see any results from that investment. That’s not why I teach workshops! It was time for management to hear a piece of my mind.

And I let them have it.

I was merciless. I told them how their current way of developing code was insanely risky—leading to deeper and more difficult problems. I explained what modern software development looks like and how the skills I had taught their developers were designed to manage risks early and often, so that software development could be safer, speedier, and far more satisfying. I described to them how critical it was to their business to embrace a new way of working that was far more conservative and effective than their old way.

Fortunately, there had been a management change during the six months

between the first and second workshops. The new management team, led by a man named Ken, was ready for change. When Ken heard what I had to say about how to build great software, it reminded him of the way he'd built hardware at a previous company he'd worked for. He immediately understood how critical it was to manage risks head-on and he trusted what I had to say.

Ken invited me to help his department make a bold change. He asked me what I suggested they do to completely overhaul their software development process, their office environment, their delivery mechanisms—everything associated with the creation of the software. I drew pictures on the whiteboard in Ken's office and explained what I considered to be ideal workspace design, how best to structure teams, and how those teams could work.

Ken loved what I told him and he wanted to make the changes immediately. Together we asked the facilities staff when they could start tearing down the cubicles and begin a redesign of the space to support collaborative software development. "In two months," was the response.

That wasn't fast enough for Ken.

So Ken asked his managers, developers, and testers to help tear down the cubicle walls *immediately*. They were happy to help, and as they started pounding on the walls, the facilities manager rushed into the room to find out where all the racket was coming from. "Oh—no, no, no," the manager yelled out, "we will do it. *Stop!*" And facilities began the work right away.

Ken was the best leader I've ever encountered during the course of my career. Our team did amazing work with Ken, transforming his organization to such a high degree of performance that a Harvard Business School professor started regularly showing up to try to understand what we'd done and how we did it.

And I'll never forget what Ken told his entire staff soon after I confronted him and his colleagues in that initial meeting—when I was fuming mad because developers weren't being allowed to use the skills I'd taught them. "We will be implementing what Josh has just explained to us," said Ken. "There is no going back to the old way. And this is not a pilot project. This is our new way of working."

And Ken meant it.

Be a bold leader who makes real change happen.



Fearless Change

Handling change rapidly, resourcefully, and gracefully is important to being agile. And it isn't easy! That's why, in 2004, Linda Rising and Mary Lynn Manns wrote an invaluable book called *Fearless Change*, and then, in 2015, added to their wisdom with another book, called *More Fearless Change*. Both books contain an anthology of useful, repeatable solutions to recurring problems called *patterns*.

Linda once gave a workshop on her *Fearless Change* patterns for a company. The CEO stopped by at the end of the day and told her how much he enjoyed the book and why he had brought Linda in to teach the workshop. The CEO explained that one of his teams was really struggling, and despite all his efforts, he found himself unable to provide the support the team needed.

Recalling a pattern he'd read in *Fearless Change* called "Do Food," the CEO went to the grocery store with his wife and asked her to help him find some kind of baked item that even he could prepare. She recommended a good brownie mix.

The next morning, the CEO got up early, baked the brownies, and took them, warm from the oven, to his struggling team. He just said, simply, "I know you have all been working hard. I wanted you to know I really appreciate your efforts, so I baked these brownies for you."

Linda shared this story with me, saying, "If the CEO had said to me that he had gotten up early to bake brownies for me, I would have felt beyond appreciated. I'm tearing up just remembering the story. I kept thinking, this is servant leadership, this is agile leadership and the greatness of humility."

One of Linda's inspirations for the "Do Food" pattern came from the following story in *Peopleware: Productive Projects and Teams*, a classic book by Tom DeMarco and Tim Lister:

In my early years as a developer, I was privileged to work on a project managed by Sharon Weinberg. She was a walking example of much of what I now think of as enlightened management. One snowy day, I dragged myself out of a sickbed to pull together our shaky system for a user demo. Sharon came in and found me propped up at the console. She disappeared and came back a few minutes later with a container of soup. After she'd poured it into me and buoyed up my spirits, I asked her how she found time for such things with all the management work she had to do. She gave me her patented grin and said, "Tom, this is management."¹

Besides "Do Food," the CEO in Linda's story implemented several other

patterns in *Fearless Change*, including Sincere Appreciation, Emotional Connection, Personal Touch, and more. These patterns helped him find a way to support the team, and not long afterward, the team resolved their struggles and started to thrive.

While change can be hard and scary, wisdom can help you navigate it fearlessly. Leverage proven patterns of change.

[¹](#). Tom DeMarco and Timothy Lister, *Peopleware: Productive Projects and Teams* (Upper Saddle River, NJ: Addison-Wesley Professional, 3rd edition, 2013).



Summary: Be Readily Resourceful

An unexpected event is an invitation to be readily resourceful. Empower your staff to be resourceful so they may exceed customer expectations. The right resourceful response to an unexpected event could be your game changer.

If a little-by-little approach doesn't make much of a difference, try the life-changing magic of going big. Unleash resourcefulness and achieve awesome outcomes via bullet-train thinking. How do you inspire people to make big improvements? Questions are the answer.

To maximize your chances of achieving outstanding outcomes, make sure you have an effective mix of deep skills.

Stop faulting yourself if you struggle to learn something. Instead, be resourceful by seeking a better explanation, book, teacher, or process. Take responsibility for your learning. Jiggle the handle.

Motivation to change is essential to making real improvements. Change does not have to equal loss. If you can change your perspective about change, you may discover how change can produce gain. Big changes aren't comfortable. It helps to get comfortable with being uncomfortable.

Learned helplessness is poison to resourcefulness. Invest time in understanding and overcoming it. Being resentful is also a clue that it's time to become resourceful.

Who's helping you learn and improve? Sometimes the best way to be resourceful is to get a coach. You can also engage the right ensemble to help you produce higher-quality results.

Focus on the long game. Learning to lose can help you win big. Let Virginia Satir's change model be your guide in successfully navigating change. If you want to become significantly better at something, be willing to become worse before you become far better. Celebrate chaos on the journey to better performance.

Make problems visible so others can see the need for change and take action. If you've got a constraint, problem, or challenge, be creative in overcoming it. Be a bold leader who makes real change happen. While change can be hard and scary, wisdom can help you navigate it fearlessly. Leverage proven patterns of change.

In Conclusion

At the beginning of this book, I recounted how I once read a few stories in Harry Beckwith's classic book *Selling the Invisible*, and how the lessons from those stories spurred me to take action. The stories I've shared in *Joy of Agility* are also meant to inspire you to take action so you can discover your own joy in agility.

Achieving your dreams is easier when you are agile. Whether you are overcoming a great challenge, handling a major unexpected event, or fulfilling a life goal, this book helps point a way forward. In whatever you seek to accomplish, ask yourself if you are being quick or hurrying, moving with ease or difficulty, in or out of balance, graceful or awkward, adapting or stuck, being resourceful or resentful.

Practice the six agile mantras daily. Learn from mistakes, from mentors, from great authors, from talented people in other industries, and from doing. Imagine your own personal Coach Wooden blowing his whistle if you are hurrying, hesitating too much, off-balance, or being insufficiently resourceful or adaptive.

Doing too many things at once is what often slows me down and interferes with my agility. Learn what commonly impedes your agility and figure out how to outsmart it.

Break on through to the agile side.

Acknowledgments

There are many people to thank in helping me create this book. I'd like to thank Maria D'Anzica, Bruce Kerievsky, Cecil Williams, Heidi Helfand, Tim Ottinger, Sandra Browne, Steve Kuo, Ashley Johnson, Matthew Carlson, Mike Rieser, Perry Reid, Julius Jervoso, Shawna Greenway, Tara Swanson, Jason Yip, Charlotte Chang, Jon Reid, Dave Rooney, Mary and Tom Poppendieck, Yves Hanouille, Mark Schell, Diana Larsen, Pat Reed, Linda Rising, Dave Nicolette, Matt Barcomb, Susan Stocker, Kevin Andrews, Lynn Langit, Jutta Eckstein, Jon Brownstein, Barton Friedland, Diane Zajak, Alexandre Freire, Ardita Karaj, and Wil Pannell.

I'd like to thank Julius Jervoso and Laura Guerin for their wonderful illustrations, Peter Economy for our conversations and his edits, Gregory Newton Brown for his suggestions and edits, the entire staff at BenBella press, and my book agent, John Willig.

I'd also like to thank the team that makes Google documents. Your product kept my work safe over the many years of writing this book. Your product allowed me to make progress on my writing, even while waiting in line, typing into a document on my mobile phone. And finally, your product helped me to easily get feedback from colleagues and friends, which helped me refine my writing.

Bibliography

Alexander, Christopher, Murray Silverstein, Shlomo Angel, Sara Ishikawa, and Denny Abrams. *The Oregon Experiment*. New York: Oxford University Press, 1975.

Avery, Christopher. *The Responsibility Process: Unlocking Your Natural Ability to Live and Lead with Power*. Pflugerville, TX: Partnerwerks Incorporated, 2016.

Baty, Gordon B. *Entrepreneurship for the Nineties*. Hoboken, NJ: Prentice Hall Direct, 1990.

Beckwith, Harry. *Selling the Invisible: A Field Guide to Modern Marketing*. New York: Warner Books, 1997.

Benson, Jim, and Tonianne DeMaria Barry. *Personal Kanban: Mapping Work | Navigating Life*. CreateSpace Independent Publishing Platform, 2011.

Berger, Jonah. *Contagious: Why Things Catch On*. New York: Simon & Schuster, 2013.

Blanchard, Ken, and Sheldon Bowles. *Raving Fans: A Revolutionary Approach to Customer Service*. New York: HarperCollins Publishers, 1993.

Bland, David J., and Alexander Osterwalder. *Testing Business Ideas: A Field Guide for Rapid Experimentation*. Hoboken, NJ: Wiley, 2019.

Blank, Steve. *The Four Steps to the Epiphany: Successful Strategies for Products That Win*. Hoboken, NJ: Wiley, 2020.

Blank, Steve, and Bob Dorf. *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch, Inc. 2012.

Catmull, Ed, and Amy Wallace. *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True Inspiration*. New York: Random House, 2014.

Crouch, Tom. *The Bishop's Boys: A Life of Wilbur and Orville Wright*. New York: W.W. Norton & Company, 2003.

DeGrandis, Dominica. *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*. Portland, OR: IT Revolution Press, 2017.

Dekker, Sydney. *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems*. Boca Raton, FL: CRC Press, 2011.

DeMarco, Tom. *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*. New York: Broadway Books, 2001.

DeMarco, Tom, and Tim Lister. *Peopleware: Productive Projects and Teams*. Upper Saddle River, NJ: Addison-Wesley Professional, 3rd edition, 2013.

Derby, Esther. *7 Rules for Positive, Productive Change: Micro Shifts, Macro Results*. Oakland, CA: Berrett-Koehler Publishers, 2019.

Duhigg, Charles. *Smarter Faster Better: The Transformative Power of Real Productivity*. New York: Random House, 2016.

Eckel, Bruce. *Thinking in Java*. Hoboken, NJ: Prentice Hall PTR, 1998.

Edmondson, Amy. *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth*. Hoboken, NJ: Wiley, 2018.

Grant, Adam. *Originals: How Non-Conformists Move the World*. New York: Penguin Books, 2017.

Gregersen, Hal. *Questions Are the Answer: A Breakthrough Approach to Your Most Vexing Problems at Work and in Life*. New York: Harper Business, 2018.

Gruver, Gary, Mike Young, and Pat Fulghum. *A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet FutureSmart Firmware*. Upper Saddle River, NJ: Addison-Wesley Professional, 2012.

Hadnagy, Christopher, and Paul Wilson. *Social Engineering: The Art of Human Hacking*. Hoboken, NJ: Wiley, 2010.

Hansen, Hardy, and Gerald M. Quinn. *Greek: An Intensive Course*. New York: Fordham University Press, 1980.

Hill, Andrew, and John Wooden. *Be Quick—But Don't Hurry: Finding Success in the Teachings of a Lifetime*. New York: Simon & Schuster, 2001.

Hollnagel, Erik, David D. Woods, and Nancy Leveson. *Resilience Engineering: Concepts and Precepts*. Boca Raton, FL: CRC Press, 2006.

Humble, Jez, Gene Kim, and Nicole Forsgren. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. Portland, OR: IT Revolution Press, 2018.

Johnson, Katherine, Joylette Hylick, and Katherine Moore. *My Remarkable Journey, A Memoir*. New York: Amistad, 2021.

Kerievsky, Joshua. *Refactoring to Patterns*. Upper Saddle River, NJ: Addison-Wesley, 2004.

Kerth, Norman L. *Project Retrospectives: A Handbook for Team Reviews*. New York: Dorset House, 2001.

Kim, Gene, Kevin Behr, and George Spafford. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. Portland, OR: IT Revolution Press, 2013.

Klein, Laura. *Build Better Products: A Modern Approach to Building*

Successful User-Centered Products. Brooklyn, NY: Rosenfeld Media, 2016.

Kondo, Marie. *The Life-Changing Magic of Tidying Up: The Japanese Art of Decluttering and Organizing*. Berkeley: Ten Speed Press, 2014.

Larsen, Diana, and Ainsley Nies. *Liftoff: Start and Sustain Successful Agile Teams*. Raleigh, NC: Pragmatic Bookshelf, 2nd edition, 2016.

Lattimore, Richmond. *The Illiad of Homer*. Chicago: University of Chicago Press, 1961.

Lencioni, Patrick. *The Five Dysfunctions of a Team: A Leadership Fable*. San Francisco: Jossey-Bass, 2002.

Lilienthal, Otto. *Birdflight as the Basis of Aviation: A Contribution Towards a System of Aviation*. Markowski International Publishers, 2018.

O'Neill, Paul, Ken Segel, and Geoff Webster. *A Playbook for Habitual Excellence: A Leader's Roadmap from the Life and Work of Paul H. O'Neill*. Independently published, 2020.

O'Reilly, Barry. *Unlearn: Let Go of Past Success to Achieve Extraordinary Results*. New York: McGraw Hill, 2018.

Patterson, Kerry, Joseph Grenny, Ron McMillan, and Al Switzler. *Crucial Conversations Tools for Talking When Stakes Are High*. New York: McGraw Hill, 2nd edition, 2011.

Patton, Jeff. *User Story Mapping: Discover the Whole Story, Build the Right Product*. Sebastopol, CA: O'Reilly Media, 2014.

Perri, Melissa. *Escaping the Build Trap: How Effective Product Management Creates Real Value*. Sebastopol, CA: O'Reilly Media, 2018.

Peters, Tom. *The Excellence Dividend: Meeting the Tech Tide with Work That Wows and Jobs That Last*. New York: Vintage, 2018.

Reason, James. *Managing the Risks of Organizational Accidents*. Burlington, VT: Ashgate, 1997.

Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business, 2011.

Rising, Linda, and Mary Lynn Manns. *Fearless Change: Patterns for Introducing New Ideas*. Upper Saddle River, NJ: Addison-Wesley, 2004.

Rising, Linda, and Mary Lynn Manns. *More Fearless Change: Strategies for Making Your Ideas Happen*. Upper Saddle River, NJ: Addison-Wesley Professional, 2015.

Ryan, Kathleen, and Daniel Oestreich. *Driving Fear Out of the Workplace: Creating the High-Trust, High-Performance Organization*. San Francisco: Jossey-Bass, 2nd edition, 2008.

Sampras, Pete, with Peter Bodo. *A Champion's Mind: Lessons from a*

Life in Tennis. New York: Crown, 1st edition, 2009.

Satir, Virginia, John Banmen, Jane Gerber, and Maria Gomori. *The Satir Model: Family Therapy and Beyond*. Palo Alto, CA: Science and Behavior Books, 2006. Sheridan, Richard. *Joy, Inc.: How We Built a Workplace People Love*. New York: Portfolio/Penguin, 2013.

Sims, Peter. *Little Bets: How Breakthrough Ideas Emerge from Small Discoveries*. New York: Free Press, 2011.

Sutherland, Jeff, and J. J. Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Crown Business, 2014.

Thomke, Stefan H. *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation*. Boston: Harvard Business Review Press, 2003.

Ward, Allen C. *Lean Product and Process Development*. Cambridge, MA: Lean Enterprise Institute, 2007.

Weinberg, Gerald M. *The Secrets of Consulting: A Guide to Giving and Getting Advice Successfully*. New York: Dorset House Publishing, 1986.

Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York: Free Press, 2nd edition, 2003. Womack, James P., Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World*. New York: Free Press, 2007.

Wooden, John, and Steve Jamison. *Wooden: A Lifetime of Observations and Reflections On and Off the Court*. New York: McGraw Hill, 1997.

Woods, David D. "Resilience is a Verb." In IRGC Resource Guide on Resilience, edited by B. D. Trump, M. -V. Florin, and I. Linkov. vol 2. Lausanne: EPFL International Risk Governance Center (IRGC), 2018.

About the Author



Joshua Kerievsky began his career as a professional programmer on Wall Street while still in university. In 1996, he founded Industrial Logic (<http://industriallogic.com>) to share the “light and magic” of software development. During the late 1990s, Industrial Logic helped pioneer agility in software development and today is considered one of the oldest and most well-respected agile consultancies in the world. Joshua and his global team of experts

love to help people and teams leverage agility in all aspects of product engineering. He and his colleagues have helped organizations including Google, GE, John Deere, Nielsen, and Ford grow high-performance teams that deliver better outcomes sooner. Joshua is a CEO, international speaker, and best-selling author. He is father to three daughters and two dogs. He loves tennis, backpacking, books, and travel. You can find him on Twitter ([@JoshuaKerievsky](#)), LinkedIn (<https://www.linkedin.com/in/joshuakerievsky/>), and the website for this book ([JoyOfAgility.com](#)).