IT Revolution
Independent Publisher Since 2013
Portland, Oregon

SOONER SAFER HAPPIER

ANTIPATTERNS AND PATTERNS FOR BUSINESS AGILITY

Jonathan Smart
with Zsolt Berend, Myles Ogilvie and Simon Rohrer

25 NW 23rd Pl, Suite 6314
Portland, OR 97210

First Edition
Printed in the United States of America
25 24 23 22 21 20 1 2 3 4 5 6 7 8 9 10

Jacket Design by Devon Smith
Book Design by Devon Smith and Tamsin Ogilvie

Library of Congress Catalog-in-Publication Data

Names: Smart, Jon, author. | Berend, Zsolt, author. | Ogilvie, Myles, author. | Rohrer, Simon, author.
Title: Sooner safer happier : anitpatterns and patterns for business agility / by Jonathan Smart with Zsolt Berend, Myles Ogilvie and Simon Rohrer.
Description: First edition. | Portland, OR : IT Revolution, [2020] | Includes bibliographical references and index.
Identifiers: LCCN 2020023325 (print) | LCCN 2020023326 (ebook) | ISBN 9781942788911 (hardcover) | ISBN 9781942788928 (epub) | ISBN 9781942788935 (kindle edition) | ISBN 9781942788942 (pdf)
Subjects: LCSH: Success in business. | Agile project management.
Classification: LCC HF5386 .S6395 2020 (print) | LCC HF5386 (ebook) | DDC 658.4—dc23
LC record available at https://lccn.loc.gov/2020023325
LC ebook record available at https://lccn.loc.gov/2020023326

ISBN: 978-1942788911
eBook ISBN: 978-1942788928
Kindle ISBN: 978-1942788935
PDF: 978-1942788942
Audio: 978-1942788959

For information about special discounts for bulk purchases or for information on booking authors for an event, please visit our website at www.ITRevolution.com.

#BVSSH

A SENSE OF URGENCY

The world changed on June 26, 2018. It happened quietly. Few people even noticed. But that was the day that, after more than a century, General Electric, the last original member of the Dow Jones Industrial Average, was removed from the index.[1]

GE was one of just a dozen firms that Charles Dow included in his list in 1896. The company appeared alongside giants such as American Tobacco, American Sugar Refining Company, and Tennessee Coal and Iron, companies that dominated what Carlota Perez, an expert on the effect of technology on socioeconomic development, calls the Age of Electricity and Engineering.[2]

As the previous age gave way to the Age of Oil and Mass Production, sugar refining lost its monopoly. Health concerns put out tobacco. General Electric, however, just kept going. It rode the change, feeding electricity into America's growing economy and transforming from an industrial conglomerate into a financial colossus. In 2004, GE was the largest firm in the world by market capitalization with a value of $382 billion.[3]

In 2016, it was one of the world's ten largest companies,[4] a symbol of how big and stable a company can become with expertise that suits the age. Just two years later, it was not. With a market value of $61 billion, only 15% of its peak

value, and with its share price contributing less than half a percent to the Dow Jones's value, GE found itself relegated from industry's top league. Something had changed.

The reasons for GE's decline are numerous (bad bets in oil, junk mortgages, and the size of GE Capital's short-term borrowing leading up to the 2008 credit crisis[5] had much to do with it); however, it's not the only large, venerable company to find it was no longer leading the pack.

The rate of creative destruction is now faster than ever. In 1964, a firm listed on the S&P 500 Index could expect to remain on the index for thirty-three years. By 2016, that tenure had fallen to twenty-four years. By 2027, companies can expect to spend no more than twelve years on the index before they're replaced. At the current churn rate, between 2018 and 2028, about half the index will have changed.[6] With companies growing and shrinking faster than ever before, there is a need to be on the right side of change in order to survive and thrive.

A look at the current set of companies listed on the Dow Jones provides one clue to the source of the dramatic turnover on the S&P 500. Alongside stalwarts such as ExxonMobil and Procter & Gamble are Verizon, Cisco, IBM, and Intel, as well as Microsoft and Apple, two of the world's largest firms by market capitalization. Currently, seven of the world's ten largest firms by market capitalization are information technology companies, including (in addition to Microsoft and Apple) Google's parent company Alphabet, Facebook, Amazon, and China's Tencent and Alibaba.[7] An economy that used to be dominated by oil and repetitive mass production has given way to one dominated by a continuous stream of information technology innovation and unique product development.

It's not just *what* the technology companies are making. What characterizes today's most highly valued organizations is *how* they make what they make. Their behavioral norms and system of work are different from anything that's come before. They are applying better approaches to work by evolving their ways of working to deliver value in a way that suits the *nature* of their work. We are in the Age of Digital.

In this new age, every company is an information technology company, whether they know it yet or not. Today, nearly all change and nearly all product development in organizations (such as a new mortgage, a new vaccine, or a new model of car) includes information technology. For example, by 2030 it is forecasted that software will account for half of the total cost of a new car.[8] The

organizations that are thriving are the ones that are leveraging information technology and treating software not as a cost center but rather as central to generating new business value.

Crucially, unlike in the age of repetitive mass production—where, for example, 1,500 cars are produced by one factory every day, one car a minute, twenty-four hours a day[9]—in the Digital Age, you don't write the same software thousands of times. Software is written *once*, rewritten a few times to improve it, and then *runs* thousands of times. Every software binary coming off the virtual assembly line is unique. People don't know what they want and you don't know how you're going to write the software until you've written it. Only once it's in the hands of people do they know what they *don't* want and do you realize how you *should* have written the code. Rather than the domain of work being repetitive, knowable, and deterministic with known-unknowns (you know how to fix it if something goes wrong), unique product development is unknowable and emergent with unknown-unknowns instead. For something that has not been done before, you don't know what you don't know until you do something and get feedback.

Over time, as compute power has increased, as we went from punch cards and valve-based computing with slow feedback loops (such as an overnight run) to microprocessors and the ability to have near-immediate feedback loops, an increasing number of software engineers realized that the then-conventional "heavyweight," sequential, stage-gate processes for software development were not optimal for the complex and emergent domain of digital knowledge work.

Practitioners felt and saw the pain. With inspiration from articles such as "The New New Product Development Game,"[10] software engineers in the late 1980s and early 1990s saw the benefits of better ways of working that were taking place in manufacturing firms like Toyota, Honda, and Xerox, with small empowered multidisciplinary teams and frequent small iterations, rather than the previous way of working with sequential, big-batch, stage-gate work passing by job role. This was in the context of product development and was heavily influenced by the legendary W. Edwards Deming, the godfather of Agile and Lean. With experimentation and experience, "lightweight" processes for software development became increasingly popular, being more suited to the emergent nature of digital work. In 2001, the values and principles of these lightweight processes were codified in the *Agile Manifesto*.

People doing product development found that these Agile principles helped them deliver value early and often with empowered teams. This led to better

outcomes. These agile ways of working—suited to unique, emergent, product development—altered everything because they correctly optimize the *approach* to the work to the *type* of work. This way of working leads to the delivery of **Better Value Sooner Safer Happier**.

Increasingly old, traditional companies—the horses rather than the unicorns—are feeling the need to exhibit agility across the whole organization in order to keep up with the "born agile" disrupters who are not held back by legacy ways of working. Organized human endeavor in the Digital Age has increasingly shifted from *repetitive* production to *unique* product development. In addition, fueled by the same technological revolution, the pace of change has become faster.

To succeed, organizations are recognizing the importance of being proficient in *ways of working* suited to and leveraging the increasingly emergent nature of work and the continuous pace of change. Organizations are recognizing a need to exhibit business agility. These ways of working are not specific to IT, nor to any sector. They are essential to survive and thrive in the Age of Digital.

Living through the Tipping Point in the Age of Digital

In order to understand the macro picture, it is helpful to look at the work of Professor Carlota Perez. In 2002, Perez wrote *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*, an analysis of the relationship between financial bubbles and technological change. In it, Perez demonstrates how, since the first industrial revolution, approximately every forty to sixty years there is a new technology-led revolution that gives rise to a paradigm shift and a new economy with societal impact. There is a recurrence of financial bubbles bursting in the middle of each technology-led revolution, caused by overinvestment in the hype, leading to a recession and then a new golden age.[11] Each recession is a tipping point from a previous normal to a new normal. Since the beginning of the dot com crash in 2000, we've been living through the tipping point in the Age of Digital.

In each age, the ways of working evolve, suited to its context. Each advances on organized human endeavor, increasing productivity. We went from factory systems to subcontracting to Taylorism and Fordism and subsequently Lean in the Age of Oil & Mass Production. Now, we are emerging into *business agility* in the Age of Digital.

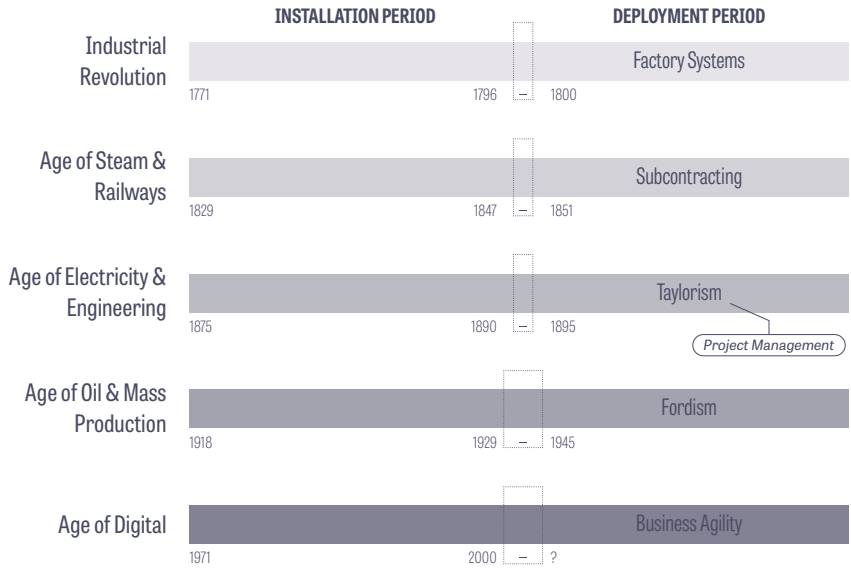

Figure A.1: Technological Revolutions

Adapted from Perez, *Technological Revolutions and Financial Capital.*

Project management and Gantt charts come from *two* technological revolutions ago, optimized for the primary context at the time, which was repetitive, knowable, deterministic, and generally physical activity. The evolution in ways of working, with time studies, reinforced the notion of managers versus workers, with a command-and-control, order-giver, order-taker, behavioral norm. Productivity improved, however, at a human cost, with workers treated as cogs in a machine.

Unfortunately, today, some organizations are still *misapplying* this way of working from more than a century ago to unique, unknowable, emergent, behavioral knowledge work: a type of activity that benefits from a wholly different approach, if there is a desire to deliver **Better Value Sooner Safer Happier**.

What This Book Is About

This book is here to help you on your unique journey to better ways of working in the Age of Digital in order to lead to better outcomes. It is a collection of

antipatterns and patterns grouped into eight chapters, each chapter being a key learning.

These are lessons learned the hard way, and they are presented here to help you avoid potential potholes. Where work is emergent there is no such thing as best practice. There is no one size fits all. Your context is unique. These antipatterns and patterns are approaches that have been experienced by the authors and observed across hundreds of organizations to—more often than not—act as a headwind (antipattern) or a tailwind (pattern) to improving outcomes. This book, and the patterns and antipatterns within it, is shared learning. As your context is unique, your mileage may vary. No other organization has the same impediments, history, and culture as your organization. There is a need to apply an emergent mindset to this emergent domain of work. Take these learnings and optimize for fast learning. Amplify experiments that work and dampen experiments that don't.

The focus of this book is on *outcomes*, not on Agile for Agile's sake, or Lean for Lean's sake. These outcomes are expressed as **Better Value Sooner Safer Happier** (or **BVSSH** for short).

Better is quality. Quality is built in rather than inspected in later. With smaller slices of value and multidisciplinary teams, changes are within a team's cognitive load (that is, complexity that fits in your head) and there is a limited "impact radius." There are fewer incidents and outages. There is less rework, less failure demand. More time is spent proactively rather than reactively.

Value is unique; it's why you are doing what you are doing. It is of value to someone. It could be financial; it could be maintaining public safety; it could be charitable.

Sooner is time to market, time to learning, to pivoting, to de-risking, to avoiding a "sunk cost fallacy," to locking in progress and value early and often.

Safer is Governance, Risk, and Compliance (GRC), information security, data privacy, regulatory compliance and resilience in chaos, be that a cyber-attack or a global pandemic. It is customers *trusting* your organization. It is agile rather than fragile. It is speed *and* control, not one or the other. It is cultural, keeping the conversation on risk alive. The better the brakes, the faster you can go.

Happier covers customers, colleagues, citizens, and climate, as it is not about "more for less" at any human or climatic cost. It is high levels of customer advocacy and colleague engagement with a positive impact to society and the one planet we live on. It is a more humane way of working.

Each of these elements balances the other. You can't force Sooner, as there will be a downward trend in Better and Happier. Improving on **BVSSH** is not prescriptive. Measures are vector metrics (trends) rather than absolutes so that improvement over time can be compared. No one is exempt from improving. Within guardrails, teams are empowered and supported to build a muscle memory of continuous improvement in line with **BVSSH**. Sometimes it's more agile and lean; sometimes it's smaller waterfalls due to history, a command and control culture of fear, and lack of psychological safety. It depends on your unique culture, history, and context.

A key theme in this book that is worth highlighting is "*Go Slower to Go Faster*." This is explored in particular in Chapters 2 and 7. There will be quick wins, and it is a fallacy to try to do too much too soon. *People have a limited velocity to unlearn and relearn*. You cannot force the pace of change in the same way that King Canute could not hold back the tide. Forcing the pace of change will likely lead to real, lasting change either not happening at all (with new labels on the same old behavior) or taking longer and with more risk. Lasting behavioral change takes as long as it takes, based on actions taken to nurture it, ignore it, force it, or sabotage it. It can be given a tailwind or a headwind. The intent is that this book gives you a tailwind.

What this book does *not* attempt to do is to touch on each body of knowledge. This is such a rich, deep, fascinating topic that it could fill a library. And if you are holding a physical copy, you are probably grateful that it doesn't weigh more! While in particular agile, lean, and DevOps are spoken of, there is no explicit attempt to cover in depth topics such as systems thinking, design thinking, user experience (UX) design, Eli Goldratt's Theory of Constraints, W. Edwards Deming's System of Profound Knowledge, and so on. It is a beautiful, never-ending learning journey, which you too can contribute to, should you wish to.

The intent is that this book will give you a tailwind for improved ways of working. This book is not just about agile. Or just about lean. It is about ways of working for better outcomes in the Age of Digital.

Who Are We?

I started out as a developer on the trading floor in investment banking in the early 1990s. I was part of a multidisciplinary, single-digit-sized team, physically sitting together, with a tribal team identity. We were the swaps trading desk.

We were all "the business" irrespective of specialism. We were naturally agile, with many-times-a-day deployments of technology-enabled value. We followed an adaptive "lightweight" process, which today would be recognized as agile and lean. It was a case of "you build it, you run it," which incentivized writing software that was supportable and resilient. It was fun and engaging.

Over time I've repeatedly taken multi-disciplinary teams on the journey from traditional to better ways of working to optimize for outcomes within the financial services industry. I have approximately thirty years of experience as an agile and lean practitioner, as a "business technologist" delivering software to generate business value, with lessons learned the hard way: by trying, sometimes succeeding, sometimes failing, learning, pivoting, and trying again.

Most recently I was leading Ways of Working at Barclays, a universal bank that dates as far back as 1690 and employs about 80,000 people globally across multiple business units. It's a large, highly regulated business. My role was to increase agility and help everyone in the bank deliver better services to Barclays' customers, release value sooner, and ensure that change was safe and compliant, with happier customers and colleagues. The benefits we saw are mentioned in the next chapter. I am currently helping organizations across industry sectors improve their ways of working, applying principles in context to optimize for **Better Value Sooner Safer Happier**.

It's teamwork, and I'm delighted that three people who have been on the same shared journey have contributed to this book. Together with Zsolt Berend, Myles Ogilvie, and Simon Rohrer—colleagues and friends—we will share in this book the most important lessons that we've been able to draw from our experiences developing and adopting better ways of working.

In the decades that Zsolt, Myles, Simon, and I have been agile and lean practitioners, we've seen the benefits that better ways of working can bring. We've repeatedly seen initiatives with a deterministic mindset, work passing by job role silos, and Think Big, Start Big, Learn Slow ways of working where hundreds of people fail at very high cost. We've also seen thousands of small, empowered teams and teams of teams improve their ways of working within guardrails, with strategic alignment, with intrinsic motivation, growing together, increasing engagement and satisfaction, regularly delivering value, and improving on **BVSSH** outcomes. We've seen how better, more humane ways of working are adopted through an organization, and we've also experienced the impediments that are a headwind for improving outcomes, and how they can be alleviated.

We've implemented many mandatory, regulatory initiatives with a fixed scope and fixed date, with a need to cease business activities if not implemented in time, all implemented with agile and lean principles, all early and all successful. In these situations, a waterfall approach would have been too risky, with late learning and back-loaded risk. Due to agility and fast time to learning, having got early learning on the least understood and the riskiest bit of these activities, we were able to go back to the UK regulator and suggest that for the benefit of global competitiveness, they might want to update the legislation, which they did. Even regulation isn't always immutable if you're learning fast enough.

Who This Book Is For

This book is aimed at leaders at all levels, in all roles, in large, complex organizations, who are on or are about to go on the journey of reinvention in the Age of Digital. You might be having some issues or you're not seeing the expected outcomes you desire or you want to set out with a guide for your journey.

This book is aimed at a broad spectrum of experience, including those who do and don't have decades of experience in agile and lean ways of working or organizational change at scale. For those who do, as per the Dunning-Kruger effect (see Chapter 3), you will realize how much you still have to learn and that you're never done learning. It is a huge pivot, how people work together, a mindset shift with a focus on outcomes over output, with finance, HR, compliance, internal audit, and real estate implications. This is a once-in-a-lifetime pivot in how human endeavor is organized. Therefore, building on all the bodies of knowledge to date, it is still a nascent topic and there is still much to learn. My intent is that this book will be updated in the future as we all continue to learn.

This book is for people organization-wide, not just IT. Nothing is out of scope if it is an impediment to delivering **Better Value Sooner Safer Happier**. Marketing, sales, legal, compliance, internal audit, HR, finance, procurement, real estate, executive committee members, non-executive directors, strategy, business units, product managers, digital, data, back office operations, the PMO, and so on should all find benefit in the antipatterns and patterns in this book.

If you are feeling that you are living through an antipattern, the intent is that this book can support you in making your case to bosses, peers, and stakeholders to help amplify ways of working that should lead to better outcomes.

How This Book Is Organized

Antipatterns and Patterns

Organizations are complex adaptive systems. There is no one way of working that suits every context. Change is emergent. Changing how you do change is emergent. Organizations are emergent, with a memory. It's emergence tripled. The only feasible way to progress is by running experiments, being sensitive to context with a fast feedback loop, and a safe-to-learn environment. There is no such thing as "best practice"; there is no one-size-fits-all set of practices that optimizes for all contexts.

What I have observed from learning the hard way and shared learning from the agile, lean, and DevOps communities is that there are common antipatterns and patterns.

An *antipattern* is a common response to a situation that, more often than not, is ineffective and risks being highly counterproductive. Antipatterns are approaches that have been seen many times to not optimize for outcomes, sometimes setting an organization back many years and creating organizational scar tissue and a strong headwind. Very occasionally an antipattern for the majority of organizations might be a pattern for one organization: for example, perhaps a scenario where cashflow is running short, and it's a high-risk, do-or-die strategy for an organization.

A *pattern* is a response to a situation that, more often than not, is effective and improves desired outcomes, of course with ups and downs, backs and forths, and swings and roundabouts, as it's all about people. A pattern can help create a tailwind. It has repeatedly improved outcomes and become "sticky." It can help create a tailwind for change. As with the antipatterns, your mileage may vary. In some rare contexts, a pattern might be an antipattern. However, I would advise caution and not use this as a rationale to knowingly adopt an antipattern with a command-and-control, deterministic mindset. I would suggest starting with the patterns and experiment with fast feedback.

This book is laid out as a series of antipatterns and their corresponding patterns. If you are reading a particular antipattern and are feeling the pain, you can look at the corresponding pattern for a suggestion on an approach that will likely generate better outcomes.

Each chapter looks at a related set of antipatterns and patterns. My aim is that you will be able to use this book as a guide on your own unique, never

ending journey. You'll be able to learn from others, avoid potholes, and accelerate while remembering that this is not a cookbook or a manual. It can't do the work for you. You should interpret the antipatterns and patterns in your own unique context. A bit like learning to ski, it's advisable to have coaching from a ski instructor, someone who can anticipate the bumps or turns ahead and help people learn to ski for themselves.

I believe these antipatterns and patterns are applicable in the context of large, bureaucratic enterprises—the economy's horses rather than its unicorns. That said, as some unicorns grow rapidly, hiring people from larger, more traditional firms, I've seen a meeting in the middle. The horses exhibit more agility and the unicorns become more bureaucratic. Even unicorns are not exempt from continuously improving. You will need to probe, sense, and respond.

Principles

With each chapter comes a number of principles. They distill the essence of the chapter to its *guiding principles*, to guide behavior and the millions of decisions that are made every day. For example, "Invite over Inflict" and "One Size Does Not Fit All." They apply across contexts.

Specific *practices* emerge by applying the *principles* to a unique context and by using coaching and experimentation, leveraging many bodies of knowledge. As Dan Terhorst-North has said: "Practices = Principles + Context."[12] The successful pattern is to identify the top ten or so principles that you feel are most important to encourage across your organization, communicate them relentlessly, and recognize behaviors in line with them. They are positive behavioral guardrails.

The intentionally long list provided in this book is intended to help you get started. The principles themselves are self-referential. You are invited to use them, and there is no one size fits all. Your context and impediments will determine which are more important to encourage.

How to Read This Book

Just as there is no one size fits all, there is no one way to read this book. It could be read left to right in a linear manner. Equally, it is intended to be "dippable." The idea is that you can hone in on an area of interest, reading the relevant antipatterns and patterns.

The book is organized into three parts.

- **Chapter 0** covers how we got here and takes a closer look at agile, lean, and DevOps.
- **Chapters 1 to 8** are the lessons learned, comprised of antipatterns and patterns.
- **Chapter 9** is advice on how to get going.

We'll now take a closer look at Chapters 1 to 8.

In Chapter 1, I start by looking at how agile and lean are not the goal and how, instead, it is important to start with why and focus on outcomes.

In Chapter 2, I talk about achieving big through small. It's not about scaling agile on top of the current bureaucracy; it's about descaling in order to scale agility and applying an agile approach to agility.

In Chapter 3, I talk about how one size does not fit all and how it is better to invite change rather than inflict it. One of the core tenets of this book is that a practice that works in one context won't necessarily optimize for outcomes in another context. Also, forcing ways of working on people is less likely to be successful than inviting participation with incentivization. Here, we look at a pragmatic approach with the VOICE acronym.

In Chapter 4, I discuss the importance of leadership, including role modeling desired behaviors, creating a psychologically safe culture where safe-to-learn experimentation is rewarded, and being less commander, more servant leader, ensuring that there is high alignment and high autonomy.

In Chapter 5, I discuss building the right thing and explain how to move from discrete output to continuous outcomes. This is the pivot from project to product, from output to outcomes.

In Chapter 6, Myles Ogilvie talks about building the thing right: how continuous compliance keeps teams on track while also keeping them free to innovate and respond. These are the minimal viable guardrails that enable safe autonomy and empowerment.

In Chapter 7, Simon Rohrer explores how continuous attention to technical excellence is essential to exhibit agility and deliver better outcomes.

In Chapter 8, Zsolt Berend explains how to become a learning organization. This is intentionally the final chapter of antipatterns and patterns, as for any organization this is an aspirational state to achieve: to become a continuously unlearning and relearning organization, in order to be the best at being better.

Throughout this book, you'll find case studies, examples, and scenarios drawn from across different industries. This is a book that, should you choose, will put you on the right side of change, create a tailwind, and help you deliver **Better Value Sooner Safer Happier**.

1 HOW WE GOT HERE

In 1992 the representatives of China's National People's Congress voted to build a hydroelectric dam at the site of the Three Gorges on the Yangtze River. Engineers drew up their plans. Work got underway. A small cofferdam was built to create a channel that would divert the flow of China's most powerful river. A new ship lock on the left bank allowed navigation to continue. Five years after the vote to start the project, the flow of the Yangtze River was blocked. A second cofferdam was used to build the dam itself together with a power station. A permanent ship lock replaced the first lock. Ten years after the vote, the second cofferdam was removed and the Yangtze River flowed again, filling the reservoir.[1]

During the construction, the Chinese government relocated more than 1.3 million people and over a thousand towns and villages were flooded. By the time the dam opened, construction had cost around $24 billion and, at its peak, employed over 26,000 workers. The dam reached as high as 185 meters and stretched more than two kilometers across the river. With twenty-six turbines, it could generate twenty times the power of the Hoover Dam. It now has thirty-two turbines and is the world's most powerful dam.[2]

It was always clear that an economy growing as quickly as China's would need new energy sources, and ideally those energy sources would need to be cleaner than coal. By implementing one stage of construction after another, China was able to successfully complete the production of one of the most complicated engineering projects ever undertaken.

While the Chinese government was building a giant dam across its most powerful river, the British government was also undertaking an ambitious project of its own: it was trying to computerize post offices so that they could improve benefit payments. The seventeen million people who then collected benefits would be given special "swipe cards." The system would reduce fraud, lower costs, and be more convenient for both government and claimants.

The card was announced in 1996. The IT project was run by the Department of Social Security (DSS) and by post office counters. Pathway, a subsidiary of International Computers Limited (ICL), won the contract to develop and install the technology. By the time the project was canceled three years later in 1999, post office counters had lost £571 million, ICL wrote off £180 million, and the DSS had laid out about £127 million. Because the system was supposed to have saved £100 million in fraudulent claims, which didn't happen, the total cost to the taxpayer of the failed project was put at about £1 billion.[3]

The cancellation of the post office benefit card project was followed by the publication of a report later that year that listed twenty-five government IT

projects that had "resulted in delay, confusion, and inconvenience to the citizen and, in many cases, poor value for money for the taxpayer." The report goes on to say that "for more than two decades, implementing IT systems successfully has proved difficult" and that "problems continue to occur in areas where recommendations have been made in the past."[4]

Those challenges aren't unique to the UK's IT initiatives. In 2013, the US government launched HealthCare.gov, a health insurance shopping site that would enable Americans to take advantage of the new Affordable Care Act. Despite the budget for the site ballooning from $93.7 million to $1.7 billion during development, it was only four days before launch that officials realized the site still had too little capacity. It crashed as soon as it opened. By the end of the day, only six of the 250,000 people who had tried to access the site were able to select an insurance plan and submit an application.[5]

In the cases above, it was a Think Big, Start Big, Learn Slow approach. The future was determined at the moment when the least was known, and there was insufficient learning until right at or after the theoretical end, which is really just the beginning. There was insufficient realization of early and often slices of value and learning. A deterministic mindset was being repeatedly applied to an emergent domain of work. And the same poor outcomes resulted.

That doesn't necessarily mean that the Chinese government is better at building things than the UK or US governments. Building a dam is knowable. There are more than 57,000 large dams worldwide.[6] China is the most dammed country in the world, with more than 23,000 large dams. Dam-building requires expertise. And having built concrete structures to hold back water 22,999 times before, those building it know what to expect, including what problems or challenges might occur. They might not be able to avoid every problem and every delay, but they know where and why delays are likely to occur. There are known-unknowns; people know what they don't know, due to having performed this activity previously many times.

Digitizing benefit payments for seventeen million people and all post offices in the UK had not been done 57,000 times before. It had *never* been done before. Building HealthCare.gov had *never* been done before. Not only that, HealthCare.gov was expected to go live on day one in the thirty-six states in the US that had declined to build their own exchanges. From zero to 250,000 users overnight in one THINK BIG, BUILD BIG, BIG BANG release.

With this work, which has never been done before, people don't know what they don't know. There are unknown-unknowns. Both of these initiatives tried

to force a deterministic-way-of-working peg into an emergent-domain-of-work hole, but that does not make it magically work. As Albert Einstein is credited with saying, "the definition of insanity is doing the same thing over and over again but expecting different results."

Instead, it is necessary to optimize for early and often learning in a real environment with real customers or consumers. This lowers the risk of delivery, generates value earlier, enables pivoting to maximize value, and locks in progress as you go. The best part is that, unlike pouring concrete, which sets, with knowledge-based products and services, such as software, this way of working is easy to do. Actually, it's the *easiest* to do.

In order to understand traditional ways of working in most large organizations today in the context of change, it's helpful to understand how we got here.

Previous Ways of Working Were Optimized for Repetitive Labor

One of the leaders of the Efficiency Movement, Frederick Winslow Taylor, did much to improve industrial processes. Working first as a machinist and then as a consultant in the 1890s, Taylor applied a scientific approach to work by using a stopwatch to analyze repetitive work, such as shoveling iron ore or inspecting ball bearings. The result, Taylorism, was a top-down, us-and-them, command-and-control management system. Workers were told when to start and stop working, managers set quotas instead of workers setting the pace of work, and tasks were increasingly specialized. Managers would watch the workers, measure their performance, and order changes. Managers planned and workers worked. Employees did what they were told. As Taylor put it:

> The work of every workman is fully planned out by the management at least one day in advance, and each man receives in most cases complete written instructions, describing in detail the task which he is to accomplish, as well as the means to be used in doing the work. This task specifies not only what is to be done but how it is to be done and the exact time allowed for doing it.[7]

While Taylor's methods increased productivity, they did little to increase happiness or satisfaction in the workplace. Indeed, it is clear that Taylor looked down

on workers: "A man who is fit to handle pig iron . . . shall be so stupid . . . that he more nearly resembles in his mental makeup the ox than any other type."[8]

Henry Gantt, the creator of Gantt charts, worked with Taylor in the early 1900s. According to Wallace Clark in *The Gantt Chart: A Working Tool of Management* (written in 1923), what we call a Gantt chart used to be called the Man Record Chart. The horizontal lines represented a worker's actual output versus what the manager (not the worker) viewed to be a reasonable quota. If you moved enough crude iron today, you could go home. If not, you had to keep working. "Long line men" were promoted and "short line men [were] very apt to do everything possible to distract the attention of others from their inferiority."[9]

The premise of the Man Record Chart was to watch over workers and follow up on perceived idleness—a continuation of the command-and-control culture of Taylorism, with managers telling workers exactly what to do. While the result was greater efficiency, it also drove a strong "us and them, managers versus workers" culture and unrest from unions.

The time study approach that Taylor championed was then built upon and improved by others, leading to the specialized production lines of Ford's Model T and eventually the pull-based, just-in-time supply methods pioneered by Toyota that now power modern automotive factories.

While Taylorism turned workers into subservient machines, advances in technology led to machines that could do the work better. The automatic loom replaced handweavers. The internal combustion engine revolutionized travel and delivery times. Telegrams and telephones increased the speed at which information could flow. The forklift and automation replaced the muscles of Taylor's steelworkers. Eventually, with the invention of the microprocessor and the arrival of the Age of Digital, labor's comparative advantage switched from following orders and moving lumps of iron to the ability to create unique products and services that deliver outcomes for customers. The means of production changed from brawn to brain.

From Repetitive Manufacturing to Unique Product Development

In Taylor's time, work was repetitive and performed by hand. Today, more and more of human endeavor is done with the head and is never the same twice, with automation taking on repetitive tasks. Today's most dynamic industrial

workplaces are no longer steel mills and fields of discarded iron. They're more likely to resemble hipster cafés with espresso machines and shared tables. In many cities, the warehouses that used to store physical goods are now trendy, bare-brick hotbeds of information technology innovation. Work has moved away from hand-making the same thing repeatedly—effort that's deterministic and has known-unknowns—to unique, knowledge-based work that is emergent and full of unknown-unknowns.

In the same way that going from the Stone Age to the Bronze Age meant not just better tools but also an entirely new society with new ways of living, organizing, and working, so the shift into today's Digital Age has produced equally large social and economic effects.

In 2011, Marc Andreessen, coauthor of the first widely used web browser (Mosaic) and cofounder of the venture capital firm Andreessen Horowitz, told *The Wall Street Journal*:

> Software is eating the world. Six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale.[10]

Organizations that have applied ways of working that suit the domain of work have not just survived, they've thrived to a degree rarely seen before. Alphabet, Amazon, Apple, and Microsoft have all been valued at over $1 trillion. Apple was the first publicly traded company to hit this landmark in August 2018, with the other three firms surpassing this valuation within eighteen months. Alphabet (Google) and Amazon went from zero to a $1 trillion valuation in just over twenty years. It is interesting to look back in time and see how for each landmark valuation there is a new normal and organizations with new ways of working that are suited to the technology revolution and type of work. The first $100 billion company was IBM in 1987, in the Age of Digital. General Motors was the first $10 billion company in 1955, in the Age of Oil & Mass Production. US Steel was the first $1 billion company in 1901, in the Age of Electricity & Engineering (and was removed from the S&P 500 Index in 2014).[11]

That doesn't mean that businesses must adopt new ways of working in this new Digital Age. Firms can choose to not adapt. A quote often attributed

to W. Edwards Deming states: "It is not necessary to change. Survival is not mandatory."

For example, in the retail apocalypse that started in 2010, approximately 10,000 stores closed in the US and 16,000 in the UK in 2019 alone prior to the COVID-19 pandemic.[12] That's five hundred stores closing every single week. The main factor cited was the shift to ecommerce. Thomas Cook, HMV, Debenhams, Bonmarche, Mothercare, Clintons, Karen Millen, Jack Wills, Bathstore, Sears, Borders, Topshop US, and Barneys are just some examples of retailers who have shut up shop or have needed to be rescued in the past few. The pandemic is accelerating the trend with as many 25,000 stores predicted to close in the US alone in 2020.[13] Meanwhile, digital natives are set to open 850 stores by 2023 in "clicks to bricks" expansion plans.[14] There are plenty of vacant stores for them to choose from.

What Are You Optimizing For?

This is an important question to ask. Within your organization, what are you optimizing for? Are you optimizing for the fast flow of safe value with high levels of customer advocacy and colleague engagement? Or for role-based silos, where work is passed over the wall to the next role-based silo with little notion of end-to-end ownership? Are you optimizing for value and time to value, or for pushing a "promise for a future solution" through endless gates and committees for years? Are you optimizing for fast learning and pivoting in order to maximize outcomes in the shortest possible time and with the least effort and least risk? Or for following a predetermined project plan with learning and risks back-loaded to the end with a large impact radius, big-bang implementation? Are you optimizing for everyone using their brains to run safe-to-learn experiments to continuously improve or for following orders?

As we've seen, organizations that have optimized their ways of working to suit the type of work have thrived. This results in higher customer expectations, raising the bar. There is a new normal, further fueled by the COVID-19 pandemic, accelerating the Age of Digital.

Given the importance of taking an optimal approach for the type of work, it is important to understand what agile, lean, DevOps, and waterfall are and their history. As this book is for leaders at all levels and in all roles in large, complex organizations, it assumes no, or little, prior knowledge of ways of working.

Certainly this is what I find in practice. People, historically, have spent very little time thinking about or improving *how* they do what they do.

What Is Agile?

Agile (along with Lean) has origins in manufacturing in Japan, heavily influenced by the teachings of W. Edwards Deming from the 1950s onwards. As we saw in "A Sense of Urgency," the 1986 *Harvard Business Review* article "The New New Product Development Game" articulates the benefits of better ways of working that were taking place in manufacturing firms like Toyota, Honda, and Xerox. The article is still remarkably up to date. These organizations, *in the context of new product development*, had small, empowered, multidisciplinary teams working in small iterations and with a clear North Star outcome. They were empowered as to *how* to achieve the mission, within guardrails, and with a high degree of experimentation. As of the date of the article, Xerox was developing new products with half the number of people and in half the time compared to the previous sequential, stage-gate process.[15]

The article uses the sport of rugby as an analogy, with the team moving together up the pitch with the ball. This led Ken Schwaber and Jeff Sutherland in the early 1990s to call their iterative and incremental approach to software product development Scrum, as per the scrum in rugby.[16]

At the same time, others in software development, myself included, were experimenting with "lightweight processes" (versus heavyweight, sequential, stage-gate processes), finding that more value was delivered sooner with less delivery risk, higher levels of engagement, and no "sunk cost fallacy." With experimentation and experience, "lightweight" processes for software development became increasingly popular, being more suited to the emergent nature of digital work. As Barry O'Reilly, author of *Unlearn*, has subsequently put it, "Think Big, Start Small, Learn Fast."[17]

In 2001, seventeen leading software developers met in Snowbird, Utah, to discuss new, lightweight methods of developing software. They produced what became known as the *Agile Manifesto*.[18] This manifesto is a set of four values and twelve principles that optimize outcomes where the type of work is unique product development. Teams who follow these principles welcome changes to requirements late in product development; trust motivated individuals to get the job done; believe that the best architectures, requirements, and designs emerge from self-organizing teams; and adjust behavior at regular intervals in

order to become more effective. While the manifesto was put together by software developers, the values and principles apply to any unique emergent type of work, not only software.

The principles laid out in the manifesto are the very opposite of the top-down management methods advanced by Taylorism. Instead of supervisors giving orders, multidisciplinary teams work together toward a clear outcome aligned to business strategy. They determine safe-to-learn experiments to test the outcome hypothesis (*probe*), measure results (*sense*), and react accordingly (*respond*). Teams are empowered within minimal viable guardrails (for example, compliance, standards, and regulation). Change and changing how you change, based on feedback loops, is essential to optimize outcomes. The principles leverage emergence to your advantage to reduce risk early and pivot to realize more value sooner.

The *Agile Manifesto* intentionally leaves it to people to figure out *how* to apply the principles because organizations are complex adaptive systems and each context is unique. It acknowledges that there is no one-size-fits-all set of practices.

One of the principles states: "Simplicity, the art of maximizing the amount of work not done, is essential." The focus is on *outcomes* over output. That is, maximizing outcomes with *minimal* output, the most value for the least effort. The definition of "productivity" is the number of units of output for each unit of input, which for unique emergent work is not optimal. Instead, the focus should be on "value-tivity," maximizing outcomes for the least output.

As we passed the tipping point in the Age of Digital, to quote Dan Mezick, an "Agile Industrial Complex" developed.[19] This is a top-down imposition of Agile practices and one-size-fits-all processes with no empowerment for teams. It is push, not pull. It is prescriptive and formulaic, not emergent or empowering, and rarely optimizes for desired outcomes in context. It is a forced infliction of emergent ways of working, done with a traditional, deterministic mindset. It is Agile snake oil, cookie-cutter Agile, Agile-in-a-box. Install it and you will be Agile. It is Agile for Agile's sake, Agile as the goal, measuring "how Agile are we." It does not necessarily lead to agility, to better outcomes.

The word "Agile" itself has collected a lot of baggage since its first inception, and I come across many people who have been burnt by an overzealous infliction of it in the past. The word generates resistance. To quote Peter Senge, author of *The Fifth Discipline*, "The harder you push against a system, the harder it pushes back."[20] To increase agility, to optimize for outcomes, given history and culture, *sometimes* the best approach is not Agile at all.

In this book, capital "A" Agile is used to refer to agility in this sense: as a noun, a product, a process, a set of practices, *doing Agile*. This alone does not necessarily translate into better outcomes. I prefer "agile" with a lower case "a," as a verb, rather than a noun, as in *being agile*, as in exhibiting *agility*. It refers to behavior, to culture, to principles, which inform millions of decisions every day. How that manifests will be unique, as your context is unique, and as we will see throughout this book.

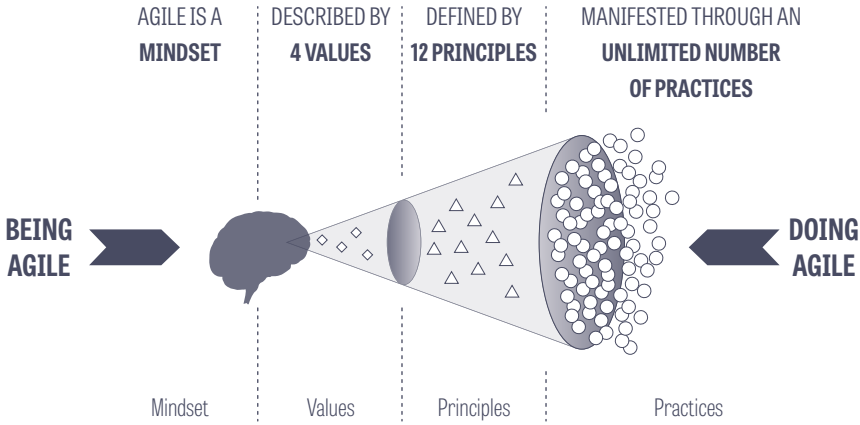

Figure 0.1: Being Agile versus Doing Agile

Adapted from Ahmed Sidky.

Sometimes, I will use the word "nimble" in place of agile in order to sense check. For example, we want to be nimble (i.e., we want to learn fast, continuously improve, and pivot), rather than we want to do Agile (we're doing standups, counting points, and doing mandated, top-down two-week sprints, but not necessarily improving and still working within a broader deterministic mindset). Equally, we don't necessarily want to "do Nimble" or run a Nimble Transformation. We *do* want to improve ways of working suited to our unique context in order to optimize for outcomes.

As organizations are complex adaptive systems, there is no one best way. The majority of agility is about behavioral norms, culture, rather than processes or tools. It's people, process, and tools, in that order.

From Mass Production to Lean Production

"Lean production" was a term coined by John Krafcick, the first American engineer hired at the Toyota-General Motors joint venture, NUMMI. His training at NUMMI included lengthy periods in Japan at Toyota factories, where he learned the fundamentals of Lean production at the source. The term first appeared in the book *The Machine That Changed the World*.[21] One description states that "Lean production is lean, because it uses less of everything compared with mass production. Half the human effort in the factory, half the manufacturing space, half the investment in tools, half the engineering hours, in half the time."[22] The core idea is to maximize customer value while minimizing waste.

Lean production began in the Toyota Production System where, due to economic necessity in the 1950s, with smaller volumes than in the US or Europe and with limited capital, Toyota's chief production engineer, Taiichi Ohno, devised a way to change machine stamping dies for body panels from a day to an astonishing three minutes. In doing so he found that it cost less per part to make small batches rather than run off enormous lots. This was because it eliminated the cost of carrying large inventories, and it meant that any stamping mistakes showed up almost immediately with minimal waste.

Building on concepts from Sakichi Toyoda (founder of Toyota) and his son, Kiichiro, Ohno also instigated a *kaizen* process of constant improvement with all workers, rather than improvement being a job role that someone else did, as was the case in the US car industry and Taylorism. Finally, Ohno developed new ways to coordinate the flow of parts within the supply chains, moving to a just-in-time, pull-based system, further eliminating costly inventory and waste and optimizing for flow. It took Ohno more than twenty years to fully implement these concepts. The result: in 2008 Toyota was the number one car manufacturer globally, stripping GM of the sales crown for the first time in seventy-eight years. Currently, Toyota has the highest market value of any automotive manufacturer, worth four times more than GM and seven times more than Ford.

In *Lean Thinking*, Daniel Jones and James Womack outline five lean principles:[23]

1. **Value**: specify value from the point of view of the customer.
2. **Value Stream**: identify the value stream and all the steps in it, from concept to cash.

3. **Flow**: limit work in progress; stabilize flow; focus on lead time, throughput, and flow efficiency; alleviate impediments to flow.
4. **Pull**: move from a push-based system of work to a pull-based system of work; go at the capacity of the system of work and don't overproduce.
5. **Perfection**: the relentless pursuit of perfection.

Lean and agile, with a common root in post–World War II Japan, have a lot in common, such as a focus on building quality in, value, flow, respect for people, a pull-based system of work, and a *kaizen* process of continuous improvement and visualizing work. However, a key area where they differ is the focus on "standardized work" in lean. Mass production looks for "good enough," and lean production looks for perfection. This is desirable for repetitive production; however, for the unique, unknowable, emergent domain of product development, "perfect is the enemy of good," to quote Voltaire. Lean production (suited to knowable, repetitive work) seeks to minimize variability, striving for perfection, in some cases targeting Six Sigma levels of perfection. Agility (suited to unknowable unique work) actively seeks and benefits from variability with multiple minimally viable, safe-to-learn experiments in order to optimize for outcomes.

What Is DevOps?

DevOps is a portmanteau that combines Development and Operations. DevOps focuses on breaking down the barriers between the teams responsible for developing a product and the teams responsible for deploying and operating the product. The term was coined by Patrick Debois when he created the DevOpsDays conference in Ghent, Belgium, in 2009. Agile in software development had alleviated the impediments to flow between customers, business analysts, developers, and testers; however, in many traditional organizations there was still a metaphorical brick wall between those building software and those running it, with a lack of shared understanding, accountability, or end-to-end flow.

Developers would build a product and then throw it over the wall at an increasing cadence, often with no notice or advice on supportability, for someone in a different role to deploy to production and support. IT Operations would tend to repetitively and manually fix issues in production without the Development team's awareness such that many issues were rarely permanently addressed. The

cost of IT Ops ("lights on") would continue to rise, squeezing discretionary spending. Typically IT build and IT run would not sit together, limiting collaboration and the ability to overhear (or even directly handle) repetitive support queries. Not surprisingly, getting closer to "you build it, you run it," sitting people together in multidisciplinary teams, automating testing and deployment, and having a focus on failure demand, supportability, resilience, and observability all lead to better outcomes. Having to support your own product is a strong motivator to maintain high quality and supportability. The primary tribal identity is aligned to the customer, the value stream, and the product(s), not the job role. The team succeeds and learns together.

In *The Unicorn Project*, Gene Kim defines five ideals of DevOps:[24]

1. **Locality and Simplicity**: alleviate dependencies between teams and components.
2. **Focus, Flow, and Joy**: the smooth flow of work that enables focus and joy.
3. **Improvement of Daily Work**: continuously improve and pay down technical debt.
4. **Psychological Safety**: a top predictor of team performance; enables improvement.
5. **Customer Focus**: optimize for customer value, not for a role-based silo.

In my experience, DevOps can have a narrow IT Dev plus IT Ops meaning and a broader enterprise DevOps meaning. The broader meaning of DevOps is delivering **Better Value Sooner Safer Happier**. It is the application of better ways of working, end to end, to deliver business and customer value, leveraging many bodies of knowledge, including agile and lean. The biggest impediment to flow, to better outcomes, might be in behavioral norms, leadership, finance, HR, PMO, real estate, governance committees, and so on. If in your context DevOps is being used in the narrow meaning, be wary of local optimization. Once the weakest link in the chain is no longer the weakest link, little value will come from continuing to strengthen it. Identify the next weakest link, which could be project-based funding for example and alleviate that, before repeating forever!

Agile, Lean, DevOps, and other bodies of knowledge are all a means to an end, not the end itself. They are shared learning in human endeavor, which can

be used in context to improve outcomes, to deliver **Better Value Sooner Safer Happier**.

What Is Waterfall?

Most large, old, traditional organizations either used to take, or still take, a waterfall approach in the context of unique change. The word "waterfall" is used as there is a sequential, stage-gate process, where work is completed at one stage before flowing to the next stage and so on. It is one-way, with big batches of work passing by job role. There is big, up-front planning and design, predicting time, cost, scope, and quality at the point when there has been the least actual learning. There is change control on the plan that inhibits agility. "Scope creep" occurs (because people are discovering the unknowable as they go) and is inhibited. Time to value is typically measured in years. The focus is on achieving a predetermined plan rather than on early and often learning to maximize value and the outcome—or to stop working on it early and move on at the lowest cost of failure.

There is sunk cost fallacy ("We've invested $100 million already. We can't write it off. Let's keep on going."). Learning is late, with a high-stakes, big-bang implementation and a large impact radius. Late learning delays the realization of value and reduces the likelihood of maximizing value. It also significantly increases delivery risk, back-loading it to when there is the least time to respond. People end up cutting corners to hit a predetermined "deadline" or feel demoralized at slipping the plan (which in reality is the gap between what is knowable and what is unknowable). The later the learning, the higher the probability of being wrong and the higher cost of being wrong. By the time something is delivered, the world has moved on. "IT doesn't move as fast as the business" is a frequent comment associated with waterfall change delivery.

Engagement is low as employees don't get to see the fruits of their labor adding value until much later, if they are lucky. People are stuck in role-based silos, with no feeling of or actual end-to-end accountability. People are promoted and incentivized within their role-based silos, leading to finger pointing. "It's not my problem. I did my bit. The hole is on their side of the boat." The problem with big-bang, waterfall failures has been described as "the application development crisis."[25]

Applying a waterfall approach in the context of unique change is a thinking error. It is miscategorizing emergent work (unknowable) as deterministic

(knowable). It is taking an approach that came about in the context of manual labor shoveling iron ore or building dam number 57,001 (tasks that have been done sufficient times before to be knowable) and applying it to unique product development (which has never been done before and is unknowable).

In the same category as waterfall is water-scrum-fall. While it is a slight improvement on a fully sequential, stage-gate process, it is not agile. It usually manifests as a waterfall project with big, up-front planning and big, up-front design, the word "sprint" ten times in the middle of the gantt chart, the work for each "sprint" having been pre-planned, and then late learning with big-bang testing and implementation. It does not exhibit agility and does not optimize for outcomes. It is still applying a deterministic mindset to an emergent domain of work.

Winston Royce, one of the first to document the waterfall sequential process, wrote in 1970 that "the implementation described is risky and invites failure."[26]

When considering the optimal approach to the type of work, it's not about agile or waterfall. It's about agile (unknowable, unique) and lean (knowable, repetitive). Waterfall is "Think Big, Start Big, Learn Slow," for which, in my opinion, there is no excuse. Why would you not optimize for early and often learning, continuous improvement, and the ability to pivot for unique change in order to de-risk and realize more value sooner and improve outcomes? Even construction has adopted agile and lean principles and practices.[27]

As we've seen, to deliver **Better Value Sooner Safer Happier**, it is important to apply the optimal approach to the work based on the type of work. In the next section, we take a look at the Cynefin framework, which is a helpful way to frame this question.

Approaching Work Based on the Domain of Work

As we've seen, product development, unique change, is *emergent*, not *deterministic*. The work is filled with unknown-unknowns and acting in the space changes the space. Conversely, a worker making wheels all day long on an assembly line knows when the wheel is built and when it's not. Likewise an organization processing ten million payment transactions a day. In that context, you want standardized work, not variability.

It's much harder when each thing you build is unique. Only once the product is built can you realize a better way of building it, or even realize that building something entirely different would better meet the needs of the con-

sumer. In an emergent domain, you want variability to learn and then amplify the experiments that optimize for the desired outcomes.

This means there is no one-size-fits-all way of working. It's not about Agile-everything or Lean-everything or DevOps-everything. It's about optimizing the way of working based on the type of work and your unique context.

The Cynefin Framework

In 1999, while working as a management consultant for IBM Global Services, Dave Snowden produced the "Cynefin" (pronounced kuh-nev-in) framework to categorize the different domains in which work today takes place. Named after the Welsh word for "habitat," the framework provides a model of five domains for problem-solving and decision-making (see Figure 0.2). It is a very useful way to determine when to take an agile approach, a lean approach, or neither.

Figure 0.2: Cynefin

Adapted from Dave Snowden.

Clear: Child's Play

The "clear" domain of the Cynefin framework is straightforward and has predictable results. This is child's play. There is no need for a project plan, a sprint,

or a backlog. A child knows that if she turns left, then right, then left again, she will arrive at school. The route is the same every day and so is the result. This domain has known-knowns and a best practice. In the UK you drive on the left. In the US you drive on the right. The relationship between cause and effect is clear. In this domain it is possible to *sense* the situation and the environment (e.g., I'm in the UK), *categorize* it based on what you know (in this country people drive on the left), and *respond* by following the rules or applying best practices (set off, driving on the left).

Complicated: Sweet Spot for Lean

The Complicated domain requires more judgment. It is knowable because this activity has been done many times before in this context. However, it's not child's play; it requires expertise. There are known-unknowns. The relationship between cause and effect needs analysis or knowledge. In this domain, you *sense*, *analyze*, and then *respond*, applying the appropriate good operating practice. There is *good* practice here, but there is no *best* practice. As it is non-trivial, there is still room for improvement, to eliminate waste, to improve quality, and to optimize flow.

For example, an IT firm installing servers in a datacenter; an automotive manufacturer building cars; an investment bank trading and processing equity trades; the HR department onboarding new employees. These activities are knowable because they've all been done many times before; however, the work requires expertise, especially when things go wrong. Even then, the failure patterns have been experienced before. This is ordered, repetitive, knowable activity. This is the sweet spot for lean.

Complex: Sweet Spot for Agile

Unique product development takes place in the Complex Domain. This is where there are unknown-unknowns and acting in the space changes the space. Cause and effect can only be deduced in retrospect. Whereas the previous two domains are *ordered*, this domain is *unordered*. There is no such thing as best practice or even good practice because activity in this domain is emergent. The best approach here is to *probe* by running a safe-to-learn experiment to test a hypothesis, to *sense* the results, and then to *respond* by amplifying or dampening the experiment.

In the Age of Digital, all software development is unique. You don't write the same code twice. People don't know what they want until they see it. You

don't know how you're going to write it until you've written it, and then it needs to be refactored as you realize how it could have been written to be more usable, maintainable, or resilient. Even installing a third-party application, such as an ERP system, is novel: that code has never been installed in that context with those data feeds, those people, and those processes before. Minimizing time to learning is key; fast feedback loops de-risk delivery and enable optimizing for outcomes. This is the sweet spot for agility.

Chaos: Act First

Sometimes decisions have to be made in a domain that is "chaotic." Knowledge here is less important than rapid action that returns order. We *act* to establish order, to stem the bleeding; *sense* where stability lies; and *respond* to turn the Chaotic into Complex. Like the Complex domain, this domain is also *unordered*.

The global COVID-19 pandemic is a good example of this domain. With mandated lockdowns in force globally and people required to stay at home, organizations scrambled remarkably quickly to act. That could have been to open up more network connections to enable huge numbers of people to work from home, or in industries such as aviation, automotive, and hospitality to shut down operations, or supermarkets and suppliers working to keep the supply chains operating. There was no time for months of planning and multiple committee-based approvals. Organizations often comment that they are at their best in these situations, with people coming together, irrespective of job role or business unit, working as one multidisciplinary team to quickly address the issue. Most then go back to their previous ways of working. Techniques stumbled upon in Chaos can end up becoming a new good or best practice in the Complicated or Clear domains for business as usual.

Confused

The last of the domains in the Cynefin framework is Confused, when it's not clear which of the domains currently apply. This can be authentic (you're really not sure, in which case break the situation down into smaller parts) or inauthentic (which means that you are complacently ignoring any distinction and carry on managing Complex situations as if they were Complicated or Clear.)

Work Moves Around Domains

Work is rarely stationary in one domain. For example, the creation of a new product, such as a new model of car, will start in the Complex domain. In

an agile manner, there will be customer focus groups, pencil sketches, computer-aided design (or "digital twin" simulation), and eventually a small-scale prototype and wind tunnel testing for quickest time and cheapest cost of learning, avoiding a sunk cost fallacy. At some point there will be a full-size prototype and eventually testing in the extremes of the Sahara and Alaska, all the time making updates to maximize the desired outcomes. Later 100,000 instances of that model are built each year, which is into the Complicated domain. Then there is a shallow dip into Chaos with a recall of certain models due to a fault, some Complex domain experimentation to fix it, and then back into Complicated domain with lean production.

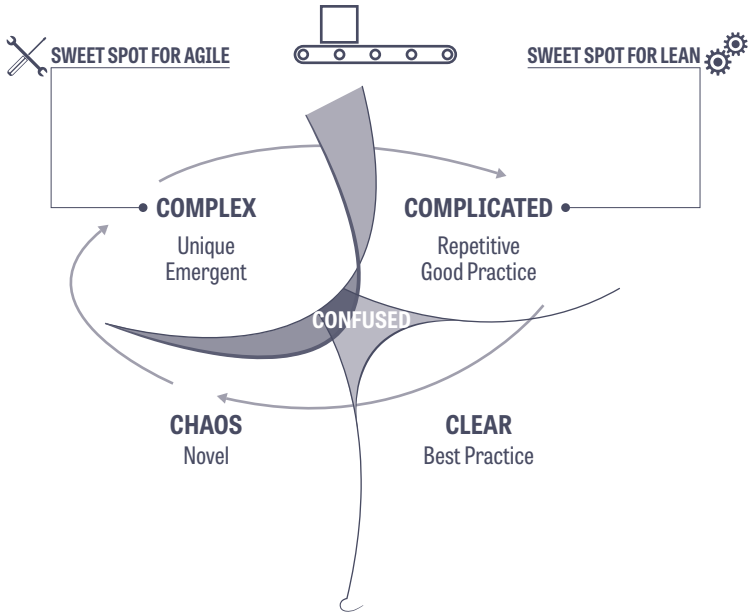

Figure 0.3: Work Moves around Domains

Software benefits from both an agile and lean approach. The software binary is agile-created and the path to production is lean, as the build, test, deploy process should run repetitively and with a high degree of automation many times a day. Periodically there will be step-change agile experimentation in the path to production and then back into lean again. Software is an agile-created box on a lean conveyor belt.

Surviving and Thriving in the Age of Digital

For those who choose to leverage the latest technological revolution and adopt ways of working that suit the nature of more of today's work, the benefits are clear. *The State of DevOps Report 2019* shows that elite performers compared to low performers deliver business value through technology 208 times more frequently, are 2,604 times faster to recover from incidents, and have a seven times lower change failure rate.[28] There is quicker learning, feedback, value, and ability to pivot to maximize desired outcomes. There is greater resilience and stability, which leads to increased satisfaction by both customers and colleagues. These factors, along with psychological safety, are positively correlated to overall organizational performance.

In my own experience, several years into being a servant leader on better ways of working across Barclays, in the context of unique product development, we saw the lead time, the time from starting work to getting it into the hands of the customer, the time to learning, to pivot, to reduce risk, reduce by two-thirds on average. The time to value and feedback was three times faster than it used to be. We saw a corresponding average 300% increase in throughput of items of value across thousands of teams for product development. The number of incidents fell by a factor of twenty, and the independently surveyed colleague engagement scores were the highest they had ever been. The teams that made the most progress reduced time to value by a factor of twenty, with throughput rising by a similar amount. Learning was twenty times faster, as was de-risking, the ability to pivot, to respond to feedback, to learn, to change direction, and to stop. Like the high performers in *The State of DevOps Report 2019*, teams were delivering **Better Value Sooner Safer Happier**.

This was after our fair share of learning the hard way and observing many Kübler-Ross curves, with peaks of excitement, troughs of disillusionment, and then, usually, climbing up to a higher point of mastery. (See Chapter 3 for more on this.) As we looked at in "A Sense of Urgency" and will explore throughout this book, lasting behavior change cannot be forced. It is not a short-term activity; it is continuous.

With the new means of production, the pace of change is getting ever faster. Change is no longer staccato, as it was in the past. Product development in particular, with a pivot from project to product, is no longer a case of big-bang builds, leaving it to go into obsolescence, letting the weeds grow, until

another big-bang slash, burn, rewrite is required. Organizations are moving to "continuous everything."

Both software systems and human systems lose information over time. Left alone they become less efficient and less maintainable. Software becomes obsolete and people, with a charitable intent, introduce bureaucracy, often with unintended consequences. The weeds grow back. Instead, we need to be tending to the garden continuously, keeping it "evergreen," nurturing culture, upgrading the plane while flying, and avoiding behavioral, process, and technical debt, which accumulates with compound interest. Change and continuous improvement should be a sustainable habit, a constant process of experimentation, feedback, learning, and pivoting to optimize for outcomes. After all, customers are not just buying a point-in-time product, they are buying ongoing innovation and an experience. And people want to work somewhere where the way of working is sustainable, engaging, and humane.

Some organizations are still using ways of working from two technological revolutions ago, misapplying them to the type of work. Others have adopted ways of working suited to the increasingly emergent nature of work in the Age of Digital. We've gone from Taylorism in the Age of Electricity & Engineering to Fordism and then Lean Production in the Age of Oil & Mass Production and now to Business Agility in the Age of Digital. Repeatedly, efficiently, sustainably, and continuously delivering **Better Value Sooner Safer Happier**. There is a new normal.

I'll talk about **BVSSH** in more detail in the next chapter; it's at the heart of this book and it should be the focus of any business that wants to survive and thrive in the Age of Digital.

ANTIPATTERN 1.1
Doing an Agile Transformation

ANTIPATTERN 1.2
Using Old Ways of Thinking to Apply New Ways of Working

1 FOCUS ON OUTCOMES: BETTER VALUE SOONER SAFER HAPPIER

PATTERN 1.1
Focus on Outcomes

PATTERN 1.2
Start with Why; Empower the How

Do you want to do or are you currently doing an Agile, Lean, or DevOps Transformation? If so, my best advice is:

Don't.

Instead, focus on the *outcomes* you want to achieve. Then you will achieve agility.

Focus on:

Better Value Sooner Safer Happier

This is the number one lesson I've learned after almost thirty years as an agile and lean practitioner delivering business value through software in the Age of Digital, from leading Ways of Working at a large, old, global, regulated organization to working with many large firms across different industry sectors. Together as a team-of-teams, as servant leaders, we experimented, learned, and pivoted.

Agile, Lean, and DevOps are not the goal. An organization can score highly on a "How Agile Are We?" test (or worse, "How Much Are We Rigidly Complying to a Specific Agile Framework?" test, or "How Many Scrum Teams Do We Have?" test) without producing better business outcomes. I've seen it happen time and time again. The wrong thing can be produced more quickly. Teams can become feature factories, a self-fulfilling prophecy of backlog replenishment with a focus on "More output!" rather than a focus on better outcomes. In addition, Agile can be viewed as an IT-only thing, no more than a local optimization, an agile bubble in a sea of traditional approaches. Or teams can exhibit cargo cult behaviors, with new labels and rituals but with the same old behaviors as before.

Agile, Lean, DevOps, design thinking, systems thinking, Theory of Constraints, and so on are all proverbial tools in a toolbox that organizations can employ to achieve desired outcomes. They are bodies of knowledge, years of wisdom acquired in the field of organized human endeavor, articulated as principles and practices. As we've seen, they are suited to specific contexts—contexts that are the new normal in the Age of Digital—as venerable old firms (the "horses" of a previous age rather than new digital "unicorns") move on from ways of working that are more than a hundred years old, originating from two technological revolutions ago in the late 1800s.

Every organization is unique and is a *complex adaptive system*. Culture change is emergent. So the interventions chosen need to be applied *uniquely in context*. There is no cookie-cutter, one-size-fits-all approach. There is no silver bullet, no snake oil, no panacea. To know whether the bodies of knowledge, the principles and practices you're using, are having the desired impact, you need to know what your desired outcomes are and keep your eye on that ball. What job are you using the bodies of knowledge for? What result do you want to produce?

At every organization I've worked for or with, those desired outcomes can be articulated as **Better Value Sooner Safer Happier (BVSSH)**.

What Is Better Value Sooner Safer Happier?

So what is **Better Value Sooner Safer Happier**? What do the terms represent and how are they measured? An important point to note is that they are *not only* IT outcomes and measures. They apply across organizations, anywhere work is being done to deliver value. They're about a collective "*our* business," not an us-and-them "*the* business," irrespective of job role. In the Age of Digital every company is a software company directly or indirectly, and there are few cases where value delivery does not in some way involve Information Technology.

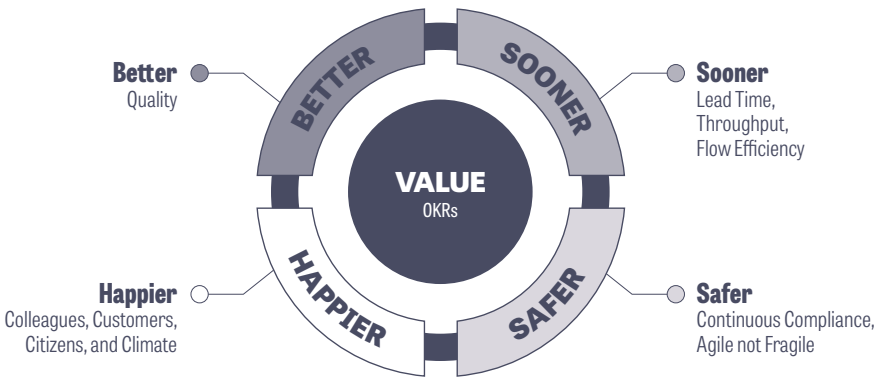

Fig 1.1: Better Value Sooner Safer Happier

Better is quality. For example, for a software product "better" could mean fewer production incidents, a faster mean time to recovery, and

improved static code analysis measures. For internal audit, "better" could be less rework of internal reports. For an operational area of an organization, such as processing payments, transactions, or loan applications, "better" could be a lower error rate. The lower the "failure demand," the lower the cost of keeping the lights on and the greater the percentage of the budget that can be spent on new value-adding activities. Quality should be built in, rather than inspected in later.

Value is in the eye of the beholder. It is unique and it is articulated via quarterly business outcome (also known as Objective & Key Results or OKRs). It's why you're in business. "Value" could be market share, revenue, units sold, P&L, margin, diversity, carbon emissions, app downloads, minutes streamed, subscribers, and so on. Value should cover the perspective of the consumer and producer.

Business outcomes are hypotheses, as we're in the emergent domain. They are nested, with a lineage up to longer-term, organization wide strategic outcome hypotheses (yearly and multi-year). There is fast feedback with daily releases of value into the hands of customers to test the hypotheses. The value measures are the KRs in OKR with leading and lagging measures. Daily, weekly, monthly nested cadences enable pivoting based on fast learning. Typically there is a monthly cadence on the quarterly business outcomes to inspect and adapt. With daily releases of value, it is possible to have daily feedback on multi-year strategic hypotheses. See Chapter 5 for more on this.

Sooner is flow, which is at the heart of agile and lean. It's about optimizing for fast and efficient flow of safe value with respect for people. There are three key measures that can be aggregated up to the organization level or disaggregated down to the team level:

- **Flow efficiency** is the percentage of time that work is actively being worked on during its elapsed end-to-end lead time, as opposed to waiting to be worked on. It is one of the most important measures, yet it is rare to find an organization that knows its flow efficiency for knowledge work. For most large service-based organizations, in my experience, flow efficiency is typically 10% or lower. This means that work is waiting at least 90% of the time. This is

where significant gains can be made. Focus on where the work *isn't*, not where the work is. Focus on the work, not the worker. The wait time is usually caused by impediments to flow, such as role-based or time-zone handoffs or multiple committee review steps, leading to work being queued. A high wait time is also caused by organizations attempting to do too much work in parallel. The more cars on the road, the slower they go. Identify and alleviate the impediments to flow and limit concurrent work in progress.

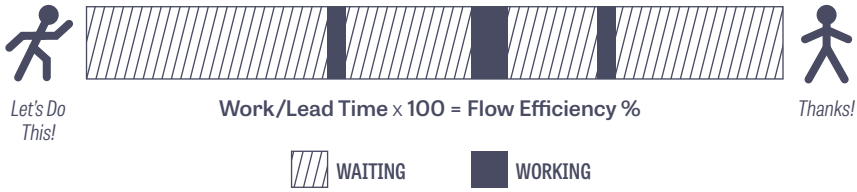

Figure 1.2 Flow (In)Efficiency

- **Lead time** is time to market, the time from starting work on an item of value to getting it into the hands of a customer. Reducing lead time enables faster feedback, quicker learning, reduced risk, earlier monetization, and the ability to pivot sooner to maximize outcomes. Lead time is a distribution—typically a Weibull distribution, a type of continuous probability distribution—that resembles a normal distribution skewed to the left and with a long tail. The recommended measure is the 85th percentile lead time and its change over time.

- **Throughput** is a count of items of value delivered into the hands of a customer in a given time period. As lead time comes down, throughput should go up. If it doesn't, then flow has an upstream impediment. Ideally, throughput should not increase directly in line with reduction in lead time. Instead, some of the time gained from reducing lead time should be used for innovation, time with customers, and continuing to improve the system of work, further

alleviating impediments to flow. We want to maximize outcomes with minimal output.

Note that the word "faster" does not appear here. "Faster" can have negative connotations. A "feature factory" can work fast, churning out features that no one wants, working harder rather than smarter.

Safer means continuous compliance, agile not fragile, a topic we cover in Chapter 6. It is about not making the news headlines due to leaking customer data. Safer is Information Security, cyber, data privacy, General Data Protection Regulation, know-your-client, anti-money laundering, fraud, and so on. It is Governance, Risk, and Compliance (GRC). Safer is speed *and* control, not choosing one at the expense of the other. Safer is cultural, with a continuous conversation on risk.

Happier is happier colleagues, customers, citizens, and climate. Improving ways of working is not at any human, societal, or climatic cost. It is about a more humane, engaging way of working, with multidisciplinary, empowered teams centered around the customer. Happier is working smarter not harder; it is improving the system of work and removing impediments. Happier is obsessing about customer satisfaction (which will lead to revenue, rather than a primary focus on short-term financial measures). Happier is also about social and climatic responsibility.

Together, **Better Value Sooner Safer Happier** balance each other. If Sooner is achieved by working people harder or cutting corners, the result will be a reduction in Better and Happier.

BVSSH contains two sets of outcomes. **Better Sooner Safer Happier** are the *how* outcomes. They measure the improvement in the system of work. **Value** is the *what*, the business outcome hypotheses that the system of work produces and that I discuss in Chapter 5. The two sets of outcomes form a virtuous circle. Improvements in the *how* leads to improvement in the *what* due to faster feedback, the ability to pivot, higher quality, and more engaged colleagues and customers.

Note that just as I don't mention "faster," I also never use the word "cheaper." A lesson learned by organizations adopting lean principles and

practices in Japan is that "cheaper" is an antipattern. It will create a headwind. People don't want to work themselves or a colleague out of a job. It is not a motivating call to action. Cheaper has negative connotations on quality and happiness.

Also, a focus on reducing visible costs often increases hidden costs via a reduction in flow efficiency. There is a hidden cost to cost-cutting. For example, introducing more handoffs, communication paths, time-zone challenges, differing incentivization, and so on all reduce flow efficiency. It increases the time that work is waiting. This reduces throughput and makes lead time longer. The system of work becomes less efficient. Learning and pivoting is slower. The company spends less, but it's also doing less and has made the system of work less effective. It's a double whammy on the ability to generate value. The organization has throttled back, both with a step change down in value production and a reduction in the gradient of adding value over time, due to reduced flow efficiency. This reduces income, which puts further challenges on cost.

Improving ways of working for product development is about "valuetivity." We want to optimize for value and time to learning. Outcomes over output. We want to maximize outcomes in the shortest time and with the least output. We want to maximize the value curve, cut the tail, and pivot to the next value curve. Typically, a focus on "cheaper" has the opposite effect, making time to value and time to learning longer.

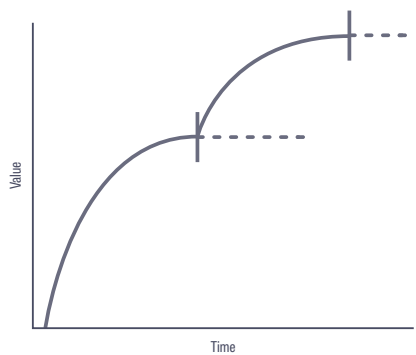

Figure 1.3: Maximize the Value Curve; Cut the Tail

Instead of cheapness, focus on **Better Value Sooner Safer Happier** outcomes and improve the system of work. As lead time reduces and throughput increases, striving for the highest value in the shortest time, with greater agility, the "income" in cost-income ratio should improve, all other things being equal and compared to maintaining the status quo.

If an organization doesn't have the runway to improve first, or macro drivers (such as a global pandemic), to change the business fundamentals and force a need to spend less and do less, my advice is to pay very close attention to the system of work. Don't increase the hidden costs with a reduction in flow efficiency and a longer time to value and learning. Don't prioritize cutting costs at the expense of flow. The result will be increased hidden costs. Do have a focus on throughput accounting as well as traditional cost accounting.

Now that you have a deeper understanding of **Better Value Sooner Safer Happier**, and you are ready to focus on outcomes rather than Agile, we will look at two of the most important, most fundamental, antipatterns. They generate a significant headwind. They are detrimental, as they do not apply an agile mindset to agility. These antipatterns (as with all antipatterns) are approaches that more often than not reduce the likelihood of achieving desired outcomes. They make a hard job harder.

ANTIPATTERN 1.1

Doing an Agile Transformation

I've met with many organizations and leadership teams that want to undertake an "Agile Transformation." The process usually begins in the same way. We sit with senior leaders and ask them *why* they want to change. The response is often silence. A couple of people will stare at the ceiling. Someone will stretch their legs. Then, eventually, one brave person will raise a hand, and say:

"It takes us too long to get new ideas to market. We're slow and inefficient."

"Good," I'll say, and write that on the board. "Anyone else?"

More replies usually come in then, and they are written down in turn:

"Everyone else is doing it."

"We're struggling to stem attrition and attract talent."

"Our customer satisfaction is trending in the wrong way."

"We don't want to be left behind."

"Beats me. I think we're doing fine as we are," says the person with their arms crossed and brow furrowed.

Each of those answers is reasonable and understandable (even the last one). We continue asking why the company might want to change how it does what it does. Eventually someone lands on the existential point: "We're being disrupted. If we don't change how we do things here, we won't survive."

Often, the organization is about to embark—or has already embarked—on an "Agile Transformation." Or they've had a bitter experience in the past with a "Lean Transformation" or Six Sigma or tried to become more "DevOps" with a focus on tooling. Typically, desired outcomes have not been articulated other than "How many Agile, Lean, DevOps teams do we have?" There are bubbles of agile in a sea of Gantt charts with predetermined solutions, dates, and spending predicted at the point of knowing the least, an annual, bottom-up financial planning process that takes six months of the year to plan and re-plan and focuses on output over outcomes. There are "drop dead dates" and "deadlines" (in *most* cases it's not life or death); RAG (red, amber, green) statuses and change control processes; a change lifecycle with twenty mandatory artifacts, most with their own stage-gate governance committee; a traditional waterfall Project Management Office; sixty-page Steering Committee decks; project plans with the word "sprint" ten times in the middle; a lack of psychological safety; a performance appraisal model that incentivizes mediocrity (underpromise to overdeliver) and uses a Think Big, Start Big, Learn Slow approach. The good news, with a charitable intent, is that the organization wants to improve.

1: Tools in the Toolbox

It's not about "Agile," or "Lean," or "DevOps" for "Agile's" or "Lean's" or "DevOps's" sake. Figuratively speaking, they are tools in a toolbox. Of course, they are much more than tools; they are behavioral principles, practices, and tools. The point is that you have a collection of bodies of knowledge to deploy optimally in context. A tool transformation, such as an "Agile Transformation," is not optimizing for outcomes, it's optimizing for the tool.

For example, let's say you want to hang more pictures on your wall, so you do a "drill transformation." To stretch the analogy further, the make of drill denotes a particular agile framework. For example, a "Bosch drill transformation" or a "Black & Decker drill transformation." At the end of your drill transformation, you may have a multidisciplinary team creating a wall full of quarter-inch holes faster, meeting hole-drilling commitments, but it

doesn't mean that the pictures are going up. Agile and lean principles, practices, methods, and frameworks are bodies of knowledge to achieve a goal, but they are not the goal. Equally, success is not defined as teams using actual tools, such as JIRA and Jenkins. A tool-led transformation does not equate to agility.

2: Cargo Cult Behavior

During World War II, America had airbases on a number of Melanesian islands in the South Pacific. Planes would land regularly, dropping off cargo such as medicine, foodstuffs, tents, and weapons that the islanders had never seen before. Once the war ended, the planes stopped coming. The islanders responded by creating what anthropologists called a "cargo cult."[1] They built wooden control towers, wore headphones made from coconuts, performed parades and ground drills with wooden rifles, and built life-sized replicas of airplanes out of straw. They had seen that when the Americans performed similar behavior, planes arrived with boxes full of goods. Despite their attempts at re-enacting this activity, the flying machines did not return to drop their cargo.

This cargo cult behavior can occur in organizations undergoing Agile Transformations. It can happen when organizations "do Agile," pursue Agile for Agile's sake, or focus on "How Agile are we?" rather than on desired outcomes. Staff might not wear coconut headphones, however, there are new role titles, iterations, standups, retrospectives, and stickies, which by themselves do not necessarily translate into better business outcomes. People are practicing the ceremonies, but the planes don't land and the cargo never arrives.

It happens at both large and small organizations. Until 2010, Nokia had the number one market share for smartphones.[2] Nokia development teams had adopted agile ways of working and, with a positive intent as an improvement on previous ways of working, would apply a "Nokia Test,"[3] which measured the company's agility in relation to the Scrum agile framework.[4] In the space of just two years, during 2011 and 2012, Nokia's Symbian operating system fell from the largest market share to extinction.[5] The last Nokia phone with the Symbian OS was unveiled in February 2012. In the UK, not a single major operator stocked it. Nokia sold its mobile phone business to Microsoft in September 2013.

In his book, *Transforming Nokia*, Risto Siilasmaa, Nokia's chairman since 2012, described how he felt when he learned that it took *two days* to compile the Symbian operating system and *two weeks* to do a complete build:

> It was as though someone had hit me in the head with a sledgehammer. . . . There were fundamental flaws in how we developed the platform that most of our profitability and near-term growth depended on. . . . As I later learned, the overall build time was two weeks! This was a recipe for catastrophe and a catastrophe was exactly what we had staring straight into our eyes.[6]

The teams could "do Agile" as much as they liked. They had Product Owners, standups, sprints, and so on, all of which are improvements on a traditional waterfall approach with a very long concept-to-cash time and a long time to learning. However, it didn't save Symbian or Nokia mobile. According to Siilasmaa, the bigger problem was a lack of psychological safety.[7] Bad news was being buried instead of exposed, discussed, and dealt with. No one had bubbled up the fact that the overall build time was two weeks. Doing Agile will not address that problem.

In my experience, the behavioral norms in an organization are the biggest lever in transforming ways of working. As I'll explain in Chapter 4, they're also the hardest to move. Had there been a primary focus at Nokia on the *outcomes* instead of on the tools, the results might have been different. A focus on Sooner might have shone a light on the long lead time to learning for Symbian features. A focus on Happier might have exposed a lack of psychological safety. Focusing on all of **BVSSH** might have kept Symbian competing with Android and iOS.

I've experienced this cargo cult behavior firsthand. It was one of my biggest lessons learned. At Barclays in 2015, I was leading an "Agile Transformation." We ran a "How Agile Are You?" self-assessment survey with four levels. The test had a positive intent, and it gave a rough indicator of who was working with agile principles and practices and who was still working with old, waterfall ways of working.

In hindsight, I wouldn't do it again. The test led to cargo cult behaviors with new labels on the same old practices. We also had teams who had adopted agile principles and methods but, for a wide range of reasons that I'll describe in later chapters, didn't produce the expected beneficial outcomes. They were focused on agile practices but not on the outcomes.

Worse still, we had targets on the four agility levels—and you get what you measure. Teams found ways to game the system, and in some cases that produced even more cargo cult behavior. Every business unit achieved their "How

Agile Are You?" targets, with some miraculous, almost unbelievable, jumps just before the end-of-year performance appraisals. The survey might have had a positive intent; however, with the benefit of hindsight, it turned out to be misguided. We learned and we pivoted.

ANTIPATTERN 1.2

Using Old Ways of Thinking to Apply New Ways of Working

Focusing on "Agile," "Lean," or "DevOps" as the end rather than the means to an end is using old ways of thinking to apply new ways of working.

A capital "A," capital "T" Agile Transformation, from the perspective of employees, infers involuntary, mandatory change being done to them, whether they like it or not. The capital "A" denotes *how* they are going to change. The capital "T" tells them they *have* to change. Both of these words carry baggage. They suggest extrinsic (push) rather than intrinsic (pull) motivation.

Capital "A" Agile in this antipattern tells employees that they *have to be* a Product Owner, Scrum Master, or Team Member. They *have to* adopt standups, retrospectives, stories, epics, and stickies. They *must* learn new jargon and become comfortable talking about velocity, story points, story mapping, planning poker, burn up, burn down, spikes, MVP, OKR, VSM, XP, CI/CD, squads, tribes, chapters, guilds, Dojo, Kata, *kaizen*, Obeya Room, and cumulative flow diagrams. It's revolution, not evolution, and you don't have a choice. Not quite as extreme as the Spanish Inquisition, it's an Agile Imposition.

Capital "T" Transformation in this antipattern represents an imposed change. It represents a program of work with a start date and an end date when the firm will have magically and permanently transformed, like a caterpillar turning into a butterfly. It is a top-down mandate, treated as a project like any other, with a deterministic mindset; big up-front planning with "deadlines" (that death analogy again); and an eighteen-month countdown. In some cases, it's a dressed-up cost-cutting exercise as a big-bang change, with a reorganization into squads and tribes, new roles, and staff layoffs. In some cases, people need to reapply for their own jobs with months of uncertainty. Are your top talent, who are able to get a job next door, likely to hang around with that degree of instability? In at least one organization that I know of, they didn't.

Typically the response in large, old organizations is a cynical: "Here we go again, yet another Transformation program. I'll sit tight, put my head in the

sand, and wait for it to blow over." I've observed some long-tenured colleagues who have developed an incredible skill and mastery in maintaining the status quo. The force is strong. For some people who are at the tail end of the job-for-life generation, and who are now in sight of their final salary pension, there is no incentive to rock the boat. Quite the opposite.

Transformation as a mandated program uses old ways of thinking to apply new ways of working. It is applying a way of working that originated in the Age of Electricity and Engineering in the late 1800s, evolved from repetitive manual labor in factories to unique emergent change in the Age of Digital. It is not applying an agile mindset to agility. To quote Martin Fowler, one of the *Agile Manifesto* signatories, "*Imposing* agile methods introduces a conflict with the values and principles that underlie agile methods."[8]

That conflict generates a number of emotional reactions, as I discuss next.

1: Fear and Resistance

Not surprisingly, mandated change and mandated *how* trigger fear and resistance. There are worries about a loss of control, uncertainty, changing habits, fear of failure, fear of incompetence, more work ("I have to do this work AND you're asking me to change how I do the work?"), change fatigue, and "better the devil you know."[9]

Managers used to traditional ways of working—to a command-and-control culture in which they give orders and see those orders carried out—fear that the change will result in a loss of control. Leaders and stakeholders accustomed to a theater of control that plans each step of a project up front (at the point of knowing the least) and assumes that the future is predictable fear embracing the reality that the domain is emergent and requires experimentation. It's easier to try to command and control the future. Leaders of role-based silos feel threatened, fearing for the empire that they've built and what that means for them. Everyone fears changing habits that they're used to and are comfortable with. Even the most confident employees can suffer from imposter syndrome. There is fear that changing a system of work that has brought them to their current position will reveal an inability, a weakness, or a vulnerability.

Other fears include concerns that the change will increase workload. With an existing need to deliver value come hell or high water, people are now being asked to change *how* that business value is delivered with jargon like "velocity,"

"sprint," and "points." Inertia plays a role too. For many people, "the devil you know" looks better than an agile and lean devil they don't know.

From an evolutionary perspective, depending on the messaging of the why and depending on how the change is approached, especially for those with a fixed mindset, change drives a fear of survival. That leads to resistance and less rational thought as the primitive brain takes over: *Can I change? What if I can't adapt? Will I still be able to pay the bills?*

As Robert Maurer explains in *One Small Step Can Change Your Life: The Kaizen Way*, the problem with the amygdala and its fight-or-flight response is that it triggers alarm bells whenever we want to make a departure from our usual, safe routines:[10]

> The brain is designed so that any new challenge or opportunity triggers some degree of fear. Whether the challenge is a new job or just meeting a new person, the amygdala alerts parts of the body to prepare for action—and our access to the cortex, the thinking part of the brain, is restricted, and sometimes shut down.[11]

2: Loss Aversion

This evolutionary fear of change is also seen in loss aversion: people's tendency to prefer avoiding losses to acquiring gains of similar value. Studies have suggested that losses are twice as powerful, psychologically, as gains.[12] The fear may be an evolutionary holdover. For our ancestors on the edge of survival, the loss of a day's food could be enough to cause starvation. The gain of additional food would be nice, but they've already survived. The extra food now needs to be stored and protected, and it's not going to add years to their life. In this scenario, the implications of loss far outweigh the benefits of gains. This evolutionary tendency to avoid losses, even to obtain gains, further cements people's desire to maintain the status quo.

3: Agentic State

Forcing change on people and dictating *how* they have to change creates extrinsic rather than intrinsic motivation. In some people this leads to an "agentic state," in which they feel compelled to obey orders, sometimes even when they think those instructions won't lead to the best or even morally right outcomes.

They pass off the responsibility for the consequences to the person giving the orders.

In 1961, Yale University psychologist Stanley Milgram began conducting a series of experiments to see whether volunteers would be willing to obey an instruction from an experimenter in a white coat to deliver high levels of electric shocks to a stranger who had answered a question wrong. In one experiment, Milgram reported that as many as 65% of the volunteers agreed to deliver what they thought was a 450-volt shock even though they feared it could be fatal.[13]

Since then, researcher Gina Perry has been through Milgram's notes and interviewed participants, publishing her findings in 2013. According to Perry, looking at the many variations of the experiment, about half the people who undertook the experiment believed that the shocks they were giving were real, and in some cases two-thirds of those refused to administer them. However, people still exhibited a state of agency, of obedience. While some people were complicit in doing something they believed was harmful, others still went through the motions, doing what they were told. They were playing their role in taking and carrying out orders from a person in perceived authority.[14]

Be wary of generating an agentic state in people. If new ways of working are imposed with an old way of behaving, as a command-and-control order, as a dictate, allowing for some national cultural differences in obedience, the majority of people will obey the order even if they don't believe it will necessarily result in a good outcome. I've come across cases where people are following orders while also wanting a change to fail. They are sabotaging it specifically by following it to the letter in order to prove their point.

A similar psychological state is "learned helplessness," where people are frozen while waiting for the next order, due to a lack of psychological safety and a command-and-control culture. If old ways of behaving and thinking are used for new ways of working, people will not think for themselves; they will not improve; they will follow orders and wait for the next one. I've seen this a number of times, with teams following mandated robotic maneuvers of agile. It did not optimize for outcomes, and it is not living the values of agile and lean.

4: Removing the Top Three Motivators: Autonomy, Mastery, Purpose

In his book *Drive: The Surprising Truth About What Motivates Us*, Daniel Pink explains that what motivates workers today, and especially workers in the

knowledge industry, isn't the promise of financial rewards.[15] Frederick Winslow Taylor's model famously used a few more dollars' salary as an incentive for manual laborers, but a couple more dollars won't motivate today's workers to move more lines of code. What will have people delivering better outcomes are *autonomy*, *mastery*, and *purpose*. We all want to feel that we control our own lives, that we're good at what we do, and that what we do matters. These are all intrinsic motivators.

When employees hear that they're undergoing an "Agile Transformation," at least two of those top three motivators are taken away. There is a lack of *autonomy* (you have to do this thing, you have no choice) and there is a lack of *mastery* (you're a beginner again, possibly after a long career).

So increasing agility can feel like a big price to pay. What are people getting for that price? Why are they being asked to pay it? If the *why* is articulated as the achievement of cost reduction or an increase in profitability—if the reason for the transformation is only to make more money for the company, perhaps also to work themselves or their colleagues out of a job—then *purpose* is also removed, taking away all three categories of human motivation: *autonomy*, *mastery*, and *purpose*.

Often, people are being forced to adopt agile or lean practices instead of being invited and incentivized to create better outcomes. As Peter Senge put it in *The Fifth Discipline*: "The harder you push, the harder the system pushes back."[16]

From Antipatterns to Patterns

Focus on Outcomes, Start with Why, Empower the How

We've seen that Agile, Lean, and DevOps are, figuratively speaking, tools in the toolbox. They are a means to an end, not the end. Doing Agile does not make you agile. A frog march of Agile leads to cargo cult behavior: the robotic maneuvers of agile are being followed, but the planes are not dropping cargo.

We've also seen that imposing Agile is not agile. It leads to fear and resistance. It is not empowering; instead, it can lead to people being in an "agentic state," complying with orders without question, thought, or ownership of the outcome. It also removes the top three motivators of work: autonomy, mastery, and purpose.

Agile and lean are bodies of knowledge, which have been accumulating for many decades, with principles, practices, ways of thinking, and behaving that,

when done well and applied in context, lead to better business outcomes. Those outcomes—and how they're balanced so that it's not at any human, societal, or climatic cost—are where the focus needs to be.

At one organization, for example, I found that adopting agile for product development was not the best way to improve outcomes. The organization had undergone multiple failed Agile Transformation attempts previously and there was history with emotional scar tissue. Instead, honoring the current roles and responsibilities and pursuing evolutionary improvement, initially with smaller, traditional waterfalls and a focus on outcomes with autonomy on the how, was the best approach to delivering **Better Value Sooner Safer Happier**. There was no revolution and thus little to fear or resist. Eventually the organization pivoted from project to product and exhibited agility with multidisciplinary teams and shorter time to market. Agile principles and practices had been adopted; however, it was done without using the A-word. Instead, it was done by people using their own brains to work out how best, in context, to improve on **BVSSH** outcomes, with support and feedback loops on **BVSSH** measures. In the process, a new and lasting muscle memory was formed, the ability of an organization to improve by itself, not waiting for an order, which is a new "learned self-helpfulness" capability. This is what you really want.

In addition to being clear on the desired outcomes and empowering *how* those outcomes are improved, there is also a need to clearly articulate the unique *why*. The organization needs to understand the reasons for being asked to improve ways of working, and those reasons need to appeal to people's intrinsic motivators. Talking about cost-cutting and layoffs will likely kill the change dead in the water. If the organization does not face an existential threat, why would someone put a lot of effort into working themselves or their colleagues out of a job? Why should they make sacrifices in order to pay institutional investors a greater dividend or meet some other short-term financial commitment made by the board? The *why* matters.

Agility needs to emerge in an agile way. Empowerment, experimentation, respect for people, self-determination, learning, everybody bringing their brains to work and continuously improving how they do what they do, are core tenets of an agile and lean mindset. The work itself is emergent and so is improving the system of work for that work.

Remember, a pattern is an approach that more often than not is successful. As every context is unique and there is no one size fits all, your mileage may vary.

PATTERN 1.1:

Focus on Outcomes

Focus on the outcomes, on **Better Value Sooner Safer Happier**, as the goal, not on Agile, Lean, or DevOps as the goal.

In his 1962 book *Diffusion of Innovations*, sociologist Everett Rogers described how innovation tended to spread first to a small number of Innovators, then reached Early Adopters, was taken up by equal numbers of Early Majority and Late Majority, before finally being used by Laggards.[17] (See Figure 1.4.)

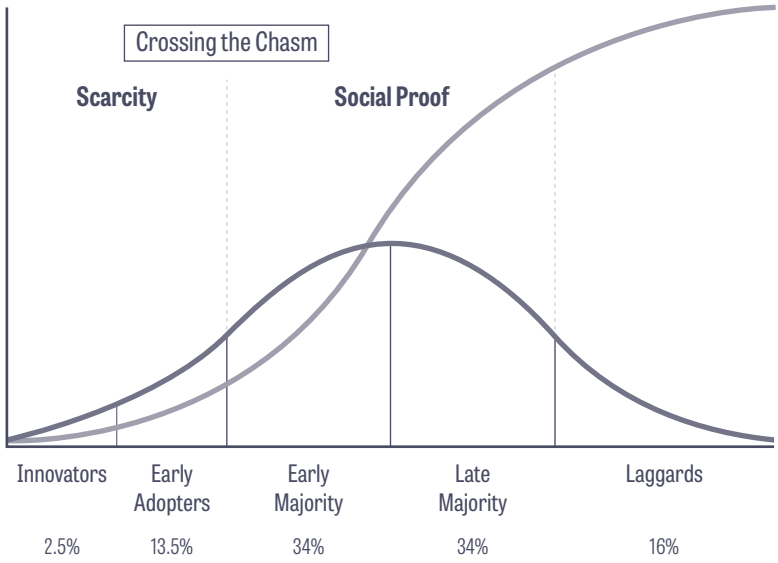

Figure 1.4: Diffusion of Innovations

As we passed the midpoint at Barclays and were getting into the Late Majority and Laggards, we recognized the need to pivot. The A-word (Agile) was an anchor, not an accelerator. It was like a magnet. The Innovators, Early Adopters, and Early Majority had one polarity. They were attracted to the new ways of working and had embraced the support the firm was providing. It helped them do what they had long been trying to do in the past. The Late Majority and the Laggards had the opposite polarity.

We pivoted to focus on outcomes. We changed our headline focus and all the visible, cultural signposts. We replaced the posters and floor-standing banners, changed the name of the team, the internal communications, and so on. We were already measuring **BVSSH** outcomes; however, we hadn't made them the headline. With the benefit of hindsight, I would have started with a headline focus on the outcomes, on **Better Value Sooner Safer Happier**.

We had previously avoided imposing a particular agile framework or approach, preferring to empower teams to decide the *how* for themselves according to their context, while we supplied support and the minimal viable guardrails that are discussed in Chapters 5 and 6. Pivoting to focus on the outcomes further increased empowerment and reduced resistance. We weren't imposing a way of working, and especially not one that might have had baggage.

My narrative when speaking to leadership teams changed from: "Hi, I'm Jon, and I'm here to make you adopt agile; you choose how." to "Hi, I'm Jon, and I'm here to help you deliver Better Value Sooner Safer Happier if you want to and if you want help."

As you can imagine, those two opening sentences generate very different responses.

No one in their right mind is not going to want to improve on **Better Value Sooner Safer Happier** outcomes. There was an incentive to improve, without mandate and without targets. The optionality at the end is full autonomy and hence intrinsic motivation. Resistance dropped away as there was nothing to resist. I've learned that the words "convince" and "resistance" should not enter the vocabulary when improving on outcomes.

It doesn't matter if you are delivering value on a mainframe with fifty-year-old code, experimenting with a new mobile app, producing an internal audit report, processing payments, or onboarding new customers, no one is exempt from choosing to improve, to be the best at being better. It's not about agile for systems of innovation and waterfall for systems of record. I find this to be an irresponsible approach. Think Big, Start Big, Learn Slow is never okay. A key part of an agile and lean mindset is continuous improvement. Irrespective of starting point and context, everyone can and should continually improve on delivering **Better Value Sooner Safer Happier**.

We made **BVSSH** outcome data transparent across business units. We showed an improvement trend (or not) over time and went from push to pull. We dropped targets and agility levels and made improving outcomes a strategic priority, looking at trends over time rather than absolute values. There

was incentivization to improve the system of work; it was no longer about agile or lean for their own sake. Having removed targets, there was empowerment in how much to improve or even to not improve at all. Making the **BVSSH** data transparent was key. It's hard data and it's hard to argue with it. And human nature means that no one wants to be the one improving the least.

The level of pull from the late majority and the laggards shot up almost overnight. After issuing the outcome data trends twice, I got a call requesting support where previous efforts had stalled. In my view, this was entirely because the focus was on the outcomes, on improving **Better Value Sooner Safer Happier**, rather than on adopting agile.

Based on this learning, from similar learnings at a range of organizations, and from case studies at conferences and sharing in the community, I've yet to find an organization where **Better Value Sooner Safer Happier** does not encapsulate the desired outcomes from better ways of working in the Age of Digital.

PATTERN 1.2:

Start with Why; Empower the How

In Antipattern 1.2, we saw how a capital "A," capital "T" Agile Transformation feels to an employee like involuntary, mandatory change being inflicted upon them, whether they like it or not. The capital "A" denotes *how* they are going to change and the capital "T" denotes that they have to change. Often organizations treat a Transformation (with a capital T) as a project with a start date and an end date, applying an old way of thinking to new ways of working. Imposing Agile is not agile, nor is treating it as a deterministic project. In addition, humans have an evolutionary bias to be averse to loss. Collectively, all of this can generate fear, resistance, and an agentic state. Forced change removes the top three motivators for people at work: *autonomy*, *mastery*, and *purpose*.[18]

In addition to being clear on the desired outcomes and empowering how those outcomes are improved, there is a need to clearly articulate the unique why for your organization in a way which appeals to intrinsic motivators. Cutting costs, increasing earnings per share, increasing return on equity, and prioritizing shareholders' short-term financial interests may not be a sufficient *purpose* for most employees. So what is?

1: Start with Why

In his book *Start with Why: How Great Leaders Inspire Everyone to Take Action*, Simon Sinek compares a company that sells what it does with a company that sells *why* it does.[19]

For example, most PC manufacturers sell computers where you select the microprocessor, the amount of memory, the size of the hard drive, and the price. The PC is a commodity and margins are low. Most PC manufacturers sell the *what*.

Apple's products, however, look like nothing else on the market. They have a design ethos, a style, and a cult following. Apple has had a strong *why* from the beginning. The company was born at a time of revolutionary, anti-establishment sentiment in Northern California. "The Apple gave an individual the power to do the same things as any company," Apple's co-founder Steve Wozniak explained to Sinek in *Start with Why*. "For the first time ever, one person could take on a corporation simply because they had the ability to use the technology."[20] Within three years Apple was a $100 million company.

In 2009, Apple's *why* was: "Everything we do, we believe in challenging the status quo, we believe in thinking differently." Today the why is: "Apple's employees are dedicated to making the best products on Earth and to leaving the world better than we found it."[21] The company has been one of the top three most highly valued, publicly traded companies every quarter since Q2 2010, and in most of those quarters it was the most highly valued company. It was the first company with a trillion-dollar market valuation and in 2019 topped *Forbes's* list of the most valuable brand for the ninth year in a row.[22]

"People don't buy what you do," Sinek says, "they buy why you do it."[23] People buy Apple's *why*.

The "buy" here could mean purchasing a product or a service, or it could equally mean buying into change. Edgar Schein, former professor at MIT's Sloan School of Management, said: "Learning only happens when survival anxiety is greater than learning anxiety. Learning anxiety comes from being afraid to try something new for fear that we will look stupid in the attempt. It can threaten our self-esteem and even our identity."[24] That anxiety is a threshold that has to be overcome in order to be willing to unlearn, relearn, and take action. If learning anxiety is higher than survival anxiety, there will be inaction. Ideally, that learning anxiety should be lowered by creating a psychologically

safe environment in which to learn, with support and coaching, rather than by increasing survival anxiety, which is what most organizations intentionally or unintentionally do.

In addition, to overcome learning anxiety, the *why*, the call to action, needs to be articulated in a form that appeals to all primary motivators. However, research has shown that the *why* for most change doesn't.[25] What the leader cares about (and typically bases at least 80% of his or her messaging on) does not tap into roughly 80% of the workforce's primary motivators for overcoming learning anxiety.

The *why* has to be about more than higher profitability, shareholder returns, or stock price. It can't only be about the company's short-term financial returns. When employees are asked what motivates them most in their work, their answers are split equally between five forms of impact:

- **Society**: They want their work to have a positive effect on society (for example, creating or protecting employment, helping those less fortunate, or improving sustainability for the benefit of our planet).
- **Customer**: They want to positively impact customer satisfaction and create brand advocates.
- **Company**: They want to have an effect on the company and its shareholders, which enables the other four forms of positive impact.
- **Team**: They want to have a positive impact on their colleagues, such as by creating a more engaging and rewarding environment or by helping team members to improve. That's an effect that people can see around them each day on people they value.
- **Individual**: They want a *why* that has a positive impact on themselves, on their levels of autonomy, purpose, and mastery, on their growth and personal development.

The pattern here is to craft a *why* for change, that people will buy and that is unique to your organization.

And then repeat that *why*. Communicate this why three more times than you think you need to and you're a third of the way done. A tip here is that when training for any way of working, whether that training is run internally or externally, have the organization's unique why at the start of every session. You cannot over-communicate the *why*. Follow it up with social proof, recognition, and reward for those who have overcome learning anxiety. Show that it's

safe for people to choose to come into the water and that others have already jumped in and are benefiting. They have an incentive to join them.

2: Improving Ways of Working Is Emergent; Empower the How

There is a need to apply new ways of thinking and behaving to new ways of working.

Change—and changing how you change—in human systems is not deterministic or reductionist. You can't take change apart, see how it works, swap a few bits, and put it all back together again. Change is emergent. The best approach to changing ways of working is not as a capital "T" Transformation, as a "project," or as a "program" with a start date and an end date. In this domain, there is no such thing as best practice, and there is no one-size-fits-all solution that optimizes outcomes for the many unique contexts in which the change takes place. It is not a case of applying a reorganization, new job titles, ceremonies, the so-called Spotify Model, and declaring that the horse has transformed into a unicorn. You'll only have glued a fake horn to the horse's head. It still won't poop rainbows.

Organizations are complex adaptive systems. The behavior of a complex adaptive system is emergent. It is not predicted by the behavior of its parts, which, for large companies, are networks of complex adaptive systems themselves. Organizations are adaptive because behavior mutates in response to change events. They look to increase what they perceive to be their survivability. Acting in the space changes the space. Any experiments that take place in a complex adaptive system are not really experiments because you can't undo them, like adding milk to coffee.

Nor is a complex adaptive system linear. It's susceptible to the butterfly effect, where a butterfly flaps its wings and there's a tornado a thousand miles away. The trick is to find the small changes that have a large *positive* effect. Complex adaptive systems have a history. They evolve and their past has a bearing on their present behavior. History and folklore are important. People don't forget. Especially if they really didn't want milk in their coffee.

As change is emergent and changing how change is done is also emergent, once you are clear on your **BVSSH** outcomes and your organization's unique *why*, the optimal approach is to apply agility to agility. As we saw in the previous chapter, this is the Complex domain of the Cynefin Framework. You have to

Think Big, Start Small, Learn Fast in your unique context. Probe, sense, and respond. Run safe-to-learn experiments with support and coaching and from within guardrails. Don't bet the farm on a big bang. At the beginning the force is strong and the corporate antibodies are powerful.

The key to success here is to invite participation. It will be the Innovators on the left of the Diffusion of Innovation Curve (Figure 1.4), that 2.5% sliver, who will volunteer to go first. They have the appetite. They are motivated by the buzz of being first and have probably been working this way formally or informally for some time. A great way to identify Innovators is to run a voluntary Community of Practice. Organize one per region, meeting in person where possible, and see who turns up every time. I previously created and chaired an internal, global, Agile Community of Practice that grew to 2,500 people. It had a three-year head start on the adoption of better ways of working firm wide, and it meant that we already knew who the Innovators and Early Adopters were across multiple business areas globally. These were people with a passionate belief derived from having been agile and lean practitioners for in some cases up to twenty years. There was social proof, in context, of how agility resulted in better business outcomes, and these people wanted to help lead the change.

Having invited the Innovators, keep the change gradient low as experiments are proven in your unique context and within risk appetite. Nail it before you scale it. Humans have a limited velocity to unlearn and relearn. You cannot force the pace of change. Seek more Innovators and Early Adopters who want to opt in from across the organization. Ensure that there are people from "our business," IT, compliance, finance, and so on, not a local optimization in one function or job role. Avoid the need to play catch-up with the Finance Department in eighteen months.

With volunteers identified, provide support. It is advisable to have a central Ways of Working Center of Enablement (WoW CoE) and coaching. (Note that it's called a "Center of Enablement," not a "Center of Excellence.") The WoW CoE is a central, small, servant leadership function for ways of working. The servant part is that it is there to support colleagues, to help mobilize the organization to remove impediments to **Better Value Sooner Safer Happier**. The leadership part is that it is there to lead the way, to shine a light, to recognize, reward, communicate, share learning, build community, and be knowledgeable on ways of working that improve on **BVSSH** outcomes. Coaching should be made available on a pull, not push basis, because like learning to ski, it's much easier if you have someone who can coach you: ski in front,

side by side, behind, then move on to the next learner. Otherwise you fall over a lot, hurt yourself, and want to go back to the hotel for a mulled wine.

Over time, as the innovators and early adopters make progress, others see the results, the recognition, the incentives, and the social proof. They join in. The **BVSSH** outcome data shows the effect that the experiments implemented so far are having. Outcome measures inch up further. More people are willing to join in. Those who were willing are now enthusiastic. Eventually, the Laggards, those who wanted a comfortable life doing what they've always done, realize that they're being left behind. When much of the company is working with new ways of working, when teams are delivering **Better Value Sooner Safer Happier**, and the Laggards have not improved when there is no reason for not improving, they stand out. That's the last thing a Laggard wants to do. In my experience, they either choose to opt out of the firm, which is fine, or they choose to join in with improving outcomes, which is preferable.

Inviting improving ways of working in order to deliver **Better Value Sooner Safer Happier** is ongoing, outcome-oriented, lower-case "t" transformation. Human systems entropy. The work never stops. You are never done improving. A former knowledge worker from Toyota in Japan once told me that all "office workers" were expected to spend 40% of their time on continuous improvement activities. I found this astounding. Two days out of every five on continuous improvement. You are never done at being the best at being better.

Case Study: Ways of Working Enablement at Nationwide Building Society

Nationwide Building Society, a large financial services organization, which is over 135 years old with more than 15 million members, is on a journey to transform ways of working. Richard James, the Ways of Working Enablement Leader, explains:

The organization recognized a need to adapt to a rapidly evolving market and keep pace with newer entrants while simplifying and reducing the cost and complexity of change—all while improving the stability and resilience of service we provide. Historically, relatively traditional in structure and approach, Nationwide was comprised of a functional set of Business communities (divisions) with a large, centralized Change team and separate IT Development and IT Operations communities leveraging a high degree of outsourcing for engineering resources. Almost all change was project-managed and delivered centrally using

a waterfall approach, with Business stakeholders engaged through business analysis for requirements documentation and subsequently developed through one or more IT Development centers with a large manual release process ahead of the transition to IT Operations. This approach was said to lack flexibility, with high costs and a slow speed to market. It needed to change.

In response to the challenge, the different communities looked at improving pace and cost through targeted change initiatives. The Technology community proposed a greater focus on agile methods, automation, in-sourcing engineering talent, and DevOps practices. The Change community put the focus on simplifying methodology and reducing handoffs between Business, Change, and IT Development. The Digital community focused on speed to market, member-centricity, and flexibility by introducing cross-functional product teams focused on customer journey re-engineering and bringing together Business, Change, and IT Development colleagues into long-lived teams. Each of these separate programmatic initiatives had deterministic plans to demonstrate marked improvements over twelve to twenty-four months.

While these separate programs were similar in intent, they overlapped in delivery, confusing colleagues and suppliers as each progressed on their respective timelines. Each initiative had a partial focus on "test and learn" but with deterministic plans of execution that sought to systematically implement improvements at an increasing pace and scale once initial learnings had been incorporated. The delivery approach for each program assumed consistent returns for all impacted areas, with all changes completed within two years of inception. While progress was made in all three programs, the net result was not as hoped. Change fatigue set in for colleagues. The board's patience eroded.

The arrival of a new COO in 2019 saw a rethink. In the first twelve months, Change, IT Development, and IT Operations were brought together alongside Controls as part of a repurposed Resilience and Agility community, with a focus on collaborating on a set of ambitions to increase pace and simplify change. A new CIO role acted as the catalyst for accelerating the move toward DevOps, alongside the introduction of a Ways of Working Center of Enablement (WoW CoE) to support teams in improving end-to-end flow while embedding a culture of continuous learning and experimentation.

The approach to ways of working pivoted from a deterministic change program to a facilitative enabling team, working at all levels, on a hypothesis-led evolution over time, focused on enabling teams to achieve Better Value Sooner Safer and Happier. Recognizing that all teams and colleagues are at different stages of their own journey, the WoW CoE doesn't seek a "one size fits all" design for agility—forming long-lived Enablement Team partnerships with teams to test, learn, and adapt while sharing emerging patterns of success across the wider organization.

With a belief in "Think Big, Start Small, Learn Fast," experimentation is the approach taken, with teams seeking to resolve impediments to flow, based on hypotheses with leading and lagging indicators aligned to outcomes, aligned to BVSSH. The benefits of descaling the evolution to fit within teams' daily working lives has been profound—tackling larger issues in context and incrementally has led to a more sustainable change where learning and experimentation are celebrated.

Summary

Want To Do a *Thing* Transformation? Don't. Focus On Outcomes.

In this chapter we've seen that the goal is not agile or lean for their own sakes. That kind of transformation can lead to cargo cult behavior and unimproved outcomes. Imposing Agile is not agile. Instead, focus on the outcomes. Focus on delivering **Better Value Sooner Safer Happier**, making use of the most appropriate proverbial tools in the toolbox in your many unique contexts, with coaching and support.

We also looked at a capital "T" Transformation and applying old ways of thinking and behaving to new ways of working. There is a mandate from the top, a traditional project or program with a deterministic mindset, a one-size-fits-all approach, and lacking a compelling why or colleague engagement. This drives fear, resistance, and can result in the removal of all of the top three motivators for people at work: autonomy, mastery, and purpose.

Instead, start with a *why* for your organization that appeals to intrinsic motivators. Focus on **BVSSH** outcomes. Invite participation and empower the how. Start with safe-to-learn experiments within guardrails and with support such as a Ways of Working Center of Enablement and coaching.

Then, keep going! Impediments are not in the path; impediments *are* the path. Take your foot off the gas and the weeds will grow back fast, so you have to keep moving. It's essential to dedicate ongoing bandwidth to support the organization in continually improving ways of working to deliver **Better Value Sooner Safer Happier**. The ultimate goal is to become a learning organization where both change and improvement is continuous. Be the best at being better.

PRINCIPLES

Focus on Outcomes

Better Value Sooner Safer Happier.

Whole Organization Agility

Agile in IT only is a local optimization.

Everything is in scope.