



WAR

Business Leadership, Technology,

PEACE

and Success in the Digital Age

IT

Excerpt

MARK SCHWARTZ

author of A Seat at the Table

Praise for *War and Peace and IT*

“*War and Peace and IT* offers a bold, insightful roadmap for building a company’s digital capacity. With the pace of change in IT accelerating at such an unprecedented rate, I consider this essential reading for my entire leadership team.”

—**François Locoh-Donou**, President & CEO, F5 Networks

“Having worked with hundreds of executives from large enterprises in my roles at AWS, it is clear to me that every CEO and CIO should read this book . . . together. As today’s leader transform their organizations for the digital era, they invariably struggle with issues of cultural change, organizational change, and rigid legacy ways of working. If only they had had this book! It is the book they need to bring together IT and the rest of their businesses in the way that can overcome those hurdles. Mark’s book is clearly informed by his executive leadership experience—both doing it himself and working with other enterprise leaders.”

—**Stephen Orban**, General Manager at Amazon Web Services
and author of *Ahead in the Cloud*

“In *War and Peace and IT*, Mark Schwartz effectively highlights how the days of silo’d functions and delivering requirements like a *War and Peace* novel to IT are over. If you and your teams aren’t out on the frontlines with IT fostering a new way of working together, your ability to succeed in the next era is likely over. If you want to learn how to embrace technology, respond effectively to ambiguity, and transform your business into an agile organization, then bring all your CXOs together and read this book with the CIO.”

—**Chris Richardson**, Chief Operating Officer, Tru Realty

“This is the book I would want with me on a walk through the woods in a Russian winter. Mark’s three books help to define how an organization should function as a whole, each approaching the question from a different angle and each as helpful in changing the organization. I am buying several copies of this book for my colleagues across all of our business operations . . . not business and IT.”

—**Josh Seckel**, Head of Agile Practice at Sevatec, Inc

“*War and Peace and IT* makes a convincing case for change: its real-life examples and the evidence it presents are concrete and compelling.”

—**Rodrigo Lobo**, Partner at PIPA Global Investments

“In an environment of chaos and uncertainty, there’s opportunity, but only if you can recognize it and react quickly. This third book in the trilogy raises the most important issue—decisions need to be made and executed in real time. Outline a set of objectives, get out of the way, and allow the creativity to flow. Mark brings the reader through this journey, and having gone through it with him at the Department of Homeland Security, I can tell you it was one of most impactful initiatives we ever undertook.”

—**Luke McCormack**, former CIO of the Department of Homeland Security

“Napoleon couldn’t centrally manage his battles in real time, but today’s leaders have no excuse. Independent cell-based teams using rapid hypothesis testing will win the battles against competitors who remain old-school. After explaining to IT leaders how to get *A Seat at the Table*, Mark Schwartz has advice for everyone else at the table.”

—**Adrian Cockcroft**, VP Cloud Architecture Strategy at AWS

WAR

Business Leadership, Technology,

PEACE

and Success in the Digital Age

IT

MARK SCHWARTZ

IT Revolution
Portland, Oregon



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

Copyright © 2019 by Mark Schwartz

All rights reserved. For information about permission to reproduce selections from this book,
write to Permissions, IT Revolution Press, LLC,
25 NW 23rd Pl, Suite 6314, Portland, OR 97210

First Edition

Printed in the United States of America

24 23 22 21 20 19 1 2 3 4 5 6 7 8 9 10

Cover and book design by Devon Smith

Author Photograph by Gary Landsman

Library of Congress Catalog-in-Publication Data

Names: Schwartz, Mark (IT manager), author.

Title: War and peace and IT : business leadership, technology, and success in
the digital age / by Mark Schwartz.

Description: Portland, OR : IT Revolution Press, [2019] |

Includes bibliographical references.

Identifiers: LCCN 2018047857 | ISBN 9781942788713 (trade pbk.) |

ISBN 9781942788720 (ePub) | ISBN 9781942788737 (kindle) |

ISBN 9781942788751 (pdf)

Subjects: LCSH: Information technology—Management. |

Technological innovations—Management. | Leadership.

Classification: LCC HD30.2 .S3878 2019 | DDC 004.068—dc23

LC record available at <https://lcn.loc.gov/2018047857>

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

War and Peace and IT

To the leaders of finance, marketing, sales, operations
I have worked with, who have taught me much:

You! Lecteur! mon semblable; mon client!

Copyrighted Material
Excerpt

Copyrighted Material
Excerpt



CONTENTS

Foreword by N.B. Himself	ix
Preface	xi
Introduction	xv
PART I PRINCIPLES	1
1 THE BUSINESS AND IT	3
2 COMPLEXITY AND UNCERTAINTY	19
3 AGILITY AND LEANNESS	31
<i>SIDE GLANCE: GRAPHS</i>	43
4 THE BUSINESS VALUE OF IT	49
PART II PARTICULARS	63
<i>SIDE GLANCE: HUMILITY AND HUBRIS</i>	65
5 THE OFF-BALANCE SHEET ASSET	73
6 RISK AND OPPORTUNITY	85
7 GOVERNANCE AND INVESTMENT MANAGEMENT	97
<i>SIDE GLANCE: INNOVATION</i>	113
8 BUREAUCRACY AND CULTURE	123
<i>SIDE GLANCE: SECURITY</i>	133
9 PLANNING AND REPORTING	141
<i>SIDE GLANCE: THE CLOUD AND THE FUTURE</i>	157
PART III PRESCRIPTIONS	165
10 ACTION PLAN	167
11 THE LEADERSHIP TEAM	181
Afterword	193
Acknowledgments	194
Bibliography	195
Notes	199
About The Author	210

ILLUSTRATIONS

Figure 1	Time versus Number of Workers	44
Figure 2	Deploys per Day per Developer	44
Figure 3	Diminishing Returns for Testing	45
Figure 4	Cost to Fix a Defect versus Time to Discovery	45
Figure 5	Risk Increases Based on Time Without a Release	46
Figure 6	The Cone of Uncertainty	46
Figure 7	Vulnerable Downloads per Month	47
Figure 8	Risk versus Frequency of Delivery	47
Figure 9	Throughput versus Batch Size	48

FOREWORD

Most Humble Readers:

I, Napoleon Bonaparte, Emperor of the French, King of Italy, Protector of the Confederation of the Rhine, First Consul of France, MBA,* do require and suggest that you read this book.

It is wrong in places, *bien sûr*, but what do you expect in the *brouillard de guerre*—this we forgive from an auteur who with insight describes how I, Napoleon, visionary and leader, brought transformation to France. Was it not I who persuaded all of France to use the metric system? I who created the Code Napoleonic?† I who dissolved the Holy Roman Empire and unified Germany? I and no other who legalized divorce, ended the Inquisition, and began the fashion for the felt bicorn hat, one of which—please note this, business readers—was sold for \$400,000 this year of 2018? And was not it I who gave my name to a kind of pastry both frosted and filled?

Do you think all of this cultural change was easy? *Non*, impediments faced on every front: the British, the Austrians, the Italians, the Russians . . . that is, I mean the weather *horrible* in Russia—if you can overcome it, you will be more successful in your transformation, I tell you this.

I commend the *auteur*, M. Schwartz, who corrects the confused mutterings of that *espèce de vache sénile*, that *tête dodelinante*, Comte Leo Tolstoy and his *mille-feuilles* of *War and Peace* filled with nonsense and frosted only with ignorance. If only I had had this book of M. Schwartz—and vast quantities of *steak*

* I conferred on Harvard Business School the privilege to award me an honorary degree.

† It is, I assert, the finest specimen of bureaucracy—which, as the auteur says in Chapter 8, is the epitome of efficiency and effectiveness.

frites and well-made boots—I would surely have triumphed in Russia, despite the weather *horrible*.

This book teaches the successful tactics innovated by me, Napoleon. How *par exemple* I brought the technological *agilité* to my forces by making mobile the artillery, so can you learn to be masterful of technology. When the *auteur* says that speed is most important—*eh bien*, M. Mack surely observed this when I took Ulm from him by moving my troops from the English Channel to the Rhine even before he finished his *café au lait*. And in regards to the motivating of troops, as M. Schwartz observes, I engaged my finance department to invest in my troops coins rather than worthless paper money.

M. Schwartz further praises focus and risk-taking. I recall to mind when I defeated the Austrians at Austerlitz by deliberately weakening my flank, which provoked them to attack it, at which moment I focused all of my forces on the very center of their position. This lesson may be useful to you.

Bon. Let us then boldly take up the banner of digital transformation and cross the Nieman River into the fray of competition and disruption. We will together make war on outdated ideas, aim our mobile artillery at the old guard, innovate to found a digital era, and make *foie gras* of the British. For as I have said in my memoirs, also required and recommended reading (*Paris: Beaudoin Frères*, 1821, available on Amazon.com), “Lead the ideas of your time and they will accompany and support you; fall behind them and they drag you along with them; oppose them and they will overwhelm you.”

A handwritten signature in black ink, appearing to read 'Napoleon', with a long, horizontal flourish underneath.

Napoleon Bonaparte
Sainte-Helene, 2018

PREFACE

May the Sheniu officials, who make the conditions of the lives of men, not cause my name to stink.

—*The Egyptian Book of the Dead*

But what does it really matter to me how So-and-so expounds his text? The main thing is that I should get some sleep.

—**Epictetus**, *Discourses*

In my role at Amazon Web Services (AWS), I meet with executives of large enterprises who are leaping to “transform” their organizations but are stumbling over cultural patterns, organizational issues, rigid processes, and implacable bureaucracy. They know that their organizations have a history of brilliant innovation, a leadership position in their markets, a passion for serving their customers . . . but somehow, despite their feeling that change is urgent, despite their worries about disruption, despite the innovation they see around them, their troops are not advancing.

I have seen a pattern to these cases. While everyone can see that digital transformation has something to do with digital technology, many don’t see that it also has to do with digital technologists—or, more precisely, the role technologists play in their companies. Ever since IT departments have existed, companies have developed ways of working with IT that actually hold the whole business back as it attempts to enter the digital age.

Having experienced both the CIO and CEO roles, I wanted to write a book about how *non-IT* leaders in the enterprise can work *with* IT to succeed in the digital world.

My last book, *A Seat at the Table: IT Leadership in the Age of Agility*, was written for IT professionals. In it I discussed recent changes in IT practice and what those changes mean for the CIO and other IT leaders. As I was writing it, I realized that these new ways of managing IT—the technique known as DevOps, in particular—were thoroughly inconsistent with the way IT has traditionally been incorporated into the broader enterprise of which it is part. At the same time, studies have shown that these new IT practices lead to vastly better business outcomes. So, I said, the CIO must change this relationship, accepting responsibility for business outcomes and taking a seat at the business strategy table.

The feedback I received from the IT community regarding the book was flattering: they loved the illustrations and my references to obscure kinds of pasta, although there were some objections to my controversial remarks on strozza-preti. But when it came to my main thesis—that the business-IT divide must be dissolved—they suggested that someone needed to inform the business community of this. Sensing another opportunity to lock myself in my room for long hours with coffee and a word processing program, I promised them that I would write something for the non-IT folks they interacted with, looking at the same questions but from the business's point of view.

Guess what? It finally hit me that I was really looking at the same problem from two sides. Enterprises are filled with technologists who are trying to bring their companies into the digital age and who are focused on achieving business value with technology. And they're frustrated trying to do so. Enterprises are also filled with non-technology business leaders who are trying to bring their companies into the digital age and achieve business value with technology. And they are just as frustrated in doing so. And here I am, a former CIO and CEO armed with a keyboard sitting between warring parties in violent agreement.

Digital transformation exposes a number of tensions that have existed within organizations. The tension between those who, according to stereotype, get pleasure from accomplishing business outcomes and those who find it in working with technology, is one. But there are also tensions between moving quickly and retaining control, between improvising and following a plan, and between

the creation of new competitive advantages and the destruction of old ones. These opposites seem impossible to reconcile; it is war, with brief periods of peace as temporary accommodations are reached.

It reminds me of something I experienced back when I was CIO at Intrax Cultural Exchange. One day my IT organization received a helpdesk ticket request from an employee that read, “Please solve Israeli–Palestinian problem!”

I was proud that they thought so highly of my IT organization that they would send us a request like that, and my team was eager to help. It only took us a moment to figure out what the ticket writer had really meant—there was a database issue that made it hard to record the biographical data of certain applicants, but no one wanted to fix it because of the sensitive politics. It was what the IT world would call a *reference data* problem. The employee just wanted us to fix the database.

I must admit that we never did fix the Israeli–Palestinian problem. But organizational oppositions I’ve described are a far more tractable problem, especially with the tools available to us today. We merely need to stop using old mental models to manage the new realities of the digital world.

Throughout history, philosophers, scientists, authors, and religious thinkers have noted tensions between opposing forces. Yin and yang in Chinese philosophy, the forces of good and evil in Zoroastrianism, creation and destruction in Hinduism, thesis and antithesis in Hegel’s dialectic. In each of these cases the great thinkers identified these tensions as the forces that drive the world forward—that account, in other words, for transformation and change.

If heads around the company are nodding yes to digital transformation, we should take advantage of all the head bobbling and get going. The route to digital wonderfulness can best be explained, I tell you, through references to Napoleon and early warfare, to Krishna driving a chariot between the Pandavas and Kauravas, to the toys with shaking heads that one finds across cultures and geographies, and to dead ancient Egyptians. You can trust me on this.

INTRODUCTION

Because technology is the negation of any definitive truth—and to dominate the transformation of things it must be this negation—the destiny of the West is radical anxiety.

—**Emanuele Severino**, *The Essence of Nihilism*

Nothing is more damaging to a new truth than an old error.

—**Goethe**, *Maxims*

Today, across industries and geographies, executives of large enterprises are struggling to transform their organizations. They have a sincere desire to “become digital” but are getting stuck along the way. They know that their enterprises are special—they have a history of brilliant innovation, a market leadership position, a mission they’re passionate about. They look at companies like Amazon and Apple, innovating at high speed, getting products to market in nanoseconds, and creating new types of business value that no one has ever before conceived. They wonder, why can’t their enterprise do the same?

It’s easy to blame cultural patterns, organizational issues, rigid processes, risk aversion, and implacable bureaucracy for their inertia. But the real obstacle is something different and harder to see. It’s the relationship between IT—the company’s organ of technical expertise—and the rest of the enterprise. The digital age demands leadership from those responsible for digital technology. But the conventional ways in which the “business” part of the enterprise and the IT part work together make that impossible.

What do enterprises want from IT? They want business outcomes. They want IT to help improve the company's competitive positioning. They want IT to drive technology-inspired innovation. They want IT to deepen relationships with customers and promote repeat business. To streamline operations and help make better use of their employees' time and mental capacity. To manage risk—particularly in information security. As the business world becomes digital, technology becomes central not just to the mechanics of running a company, but to the company's ability to compete and survive.

What do enterprises actually ask and incentivize IT to do? To deliver projects on time and according to plan. To reduce IT costs. To provide good customer service to the company's own employees. To take in requirements from the business units and deliver IT capabilities to fulfill them precisely as stated. To control risk by not allowing scope to creep and managing to deadlines.

What we *ask* of IT isn't what we *want* of IT; it's at best dimly related.

As the cliché has it, the pace of change is accelerating. That's nothing new to technologists—for decades, changes in technology have come faster and faster as startlingly inventive people join the technology world; as software, hardware, and IT service providers pour resources into growing their markets; and as demand for features, scale, security, and resilience drive innovation. A software developer, for example, must constantly learn about and experiment with new programming languages, new practices, and new mental models for architecting and designing IT systems.

The increased pace now applies to business and product strategy as well, largely because technology is so much at the center of everything an enterprise does. Why are executives on email twenty-four hours a day? It's not just because email is available; it's because they actually feel a need to make decisions while eating pasta, watching reality TV, and emerging from REM sleep. We feel that urgency in our stress levels. Lowered barriers to entry let disruptive companies transform industries in the flick of an eyelid. The cloud, the pliability of well-written software, the unmediated access to customers globally over the internet, the availability of venture capital, an incoming workforce that has an intuitive understanding of younger markets—all of

these raise the risk that a company that was once comfortable will wake up one day to find itself dead or delisted, or at least no longer welcome on the S&P 500 index.

It's not just startups that threaten established enterprises. Other traditional enterprises have found ways to draw on the magic of today's technology to pull new products out of a proverbial hat, to make costs disappear, or to transform red financials to black. According to a Boston Consulting Group (BCG) study, companies are dropping from top-three positions in their industries faster than ever before, and once they do, are also likely to drop from the top ten rankings within five years.¹

There is simply no way an enterprise can feel comfortable with its status quo, or move slowly and tentatively toward a vaguely glimpsed future; urgency and clarity are the demands of the day. Forty-seven percent of CEOs report feeling pressure from their boards to digitally transform²—whatever they might mean by that—and sixty-two percent already have some such transformation underway.*³ Companies that move too slowly are destroying business value every nanosecond.

In their book *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*, authors Nicole Forsgren, Jez Humble, and Gene Kim identify four areas where organizations must accelerate:

- delivery of goods and services to delight their customers
- engagement with the market to detect and understand customer demand
- anticipation of compliance and regulatory changes that impact their systems
- response to potential risks, such as security threats or changes in the economy⁴

To accelerate, enterprises must find a way to bring technology to the heart of their work, for just as technology is causing this disruption, it is technology that provides the solution. It's the internet that lets them quickly reach customers across the world, it's the cloud that lets them instantly acquire the

* Also 42% say “digital first” or “digital to the core” are their company's default digital postures.

infrastructure they need, and it's the changeability of software that lets them continuously innovate and transform to meet emerging demands.

Ironically, enterprises often consider IT to be a hindrance—a frictional force that slows them down as they grind forward to deliver value for their customers. But it's IT departments, which have lived with change and uncertainty for their entire existence, that have developed ways of coping with the constant pressure to adapt. The IT folks have quietly been streamlining their processes and finding ways to figure skate delicately at high velocity; what remains is for enterprises to put on their skates and get comfortable gliding on the digital world's slippery surfaces.

Leaders of digital transformations often look around their organizations and see heads nodding. Everyone seems to agree that change is needed to survive in the digital age. Everyone understands that it's urgent and that there is a risk of being disrupted if the company doesn't transform quickly enough. Capital markets are demanding growth and innovation, while the board wants management to invest in becoming future-ready. Executives are aware that competitors are learning how to build closer digital relationships with customers. Frankly, it's rare to so easily arrive at consensus.

But nothing seems to be happening. Heads are bobbling yes-yes-yes, plans are being discussed, priorities are being set . . . but the digital prince remains an analog frog. It's easy to blame the lack of progress on corporate culture, a lack of up-to-date skills among employees, rigid bureaucratic processes, lack of cohesion across business silos, heavy compliance requirements, accounting rules, or inflexible auditors. For any large enterprise, those are indeed important factors. But many of them are within the company's control and others, as I'll show later, are outputs of successful transformation—not prerequisites.

If you want to unlock your enterprise's digital transformation, you must change not only its relationship with technology, but its relationship with its technologists. Conventional wisdom has settled on a way of integrating IT into the enterprise that hasn't been very effective up to now and remains much less likely to be effective in the digital future. IT and the business face each other across a daunting chasm of stereotypes and perceived risk like rows of bobblehead dolls, bobbling and smiling at each other with goodwill and mutual

respect, coupled with a formality that precludes intimacy. The key to digital transformation is to change the way IT and the business interact.

Over the decades that IT has been part of the corporate landscape, it's been regarded as a sort of arms-length contractor serving the rest of the business. A business unit decides what IT capabilities it needs, writes a requirements document, negotiates an understanding with IT about scheduling and costs, then tosses its requirements over the wall for delivery. IT is then responsible for fulfillment, delivering what was requested on the schedule it agreed to. We speak of "IT and the business" as if we're referring to two different things, and we encourage IT to treat the business as its customer. It's as though IT were an outside service provider full of people who just happen to be employees of the same company.

Digital transformation, on the contrary, means making technology central to the way an enterprise defines itself, rather than a utility or support function that can just as easily be outsourced. But this can only happen if the technologists are as much a part of the business as employees in marketing, finance, and operations.

Changing this relationship can be uncomfortable for both sides. On one hand, business employees have gotten used to being treated as IT's customers, whether the customer service they received was tip-worthy or not. This contractor-like model has given the illusion of control to the business—the feeling that even when they don't understand the technological details, they can at least hold IT accountable to some performance standard. They can feel like they've shifted the burden of technical uncertainty, complexity, and change onto the IT folks, and thereby gained predictability and simplicity for themselves. As long as IT said a project would be completed by a certain date, uncertainty had been managed away, or at least could be overseen by way of conventional risk management practices.

On the other hand, IT departments have never had to take responsibility for business outcomes. Someone else always decides which technology capabilities will create business value; someone else works to harvest the business value from the products IT delivers. IT has been able to say, "We can't do anything until we get your requirements," while enforcing policies and standards that might constrain business operations. By pushing the burden of value

determination to the business, IT can feel like it's free of the biggest uncertainties and complexities in its activities.

As we move into today's digital world, uncertainties and complexities are becoming an everyday matter for everyone—IT and non-IT alike. We can no longer separate technology risk from business risk, or technology opportunity from business opportunity. The business must accept the risk and uncertainty that comes with technology, while IT must accept the risk and uncertainty that comes with business.

It's not just IT that finds itself distanced from the core strategic activity of the enterprise—there is a deeper and more general issue at play. As business and technical functions became more complex and specialized, organizations came to structure themselves into functional silos. Finance was expected to focus on finance, marketing on marketing, and IT on IT. Each area was assigned goals specific to its functions, which were then further subdivided and passed down to subspecialty areas. In this way, the reasoning went, each functional area could be held more accountable for things that no one outside completely understood anymore. But organizations are now paying the price for this fracturing as they try to develop a coherent strategic approach to the digital world.

The chief financial officer (CFO), for example, has often wound up focused on cost reduction and the operational efforts of seeing that the books get closed on time. According to a McKinsey study, two-thirds of CFOs think they should spend less time on traditional finance activities and more on strategic leadership.⁵ About 30% of the finance department's effort is invested just in the mechanics of assembling data and resolving inconsistencies.⁶

The digital world, however, demands that the CFO play more of the role of strategic business advisor—the custodian of shareholder value or mission delivery.⁷ In a digital organization the CFO drives competitive advantage by applying capital to opportunities as they arise, turning data into actionable business insights, and managing risk strategically. In place of cost reduction, the digital CFO focuses on making processes leaner, thereby removing waste and increasing the enterprise's velocity.

Among chief marketing officers (CMOs), the story is similar: 74% say their role doesn't allow them to have the impact on the business that they should.⁸ Today, marketing must handle more countries, more customer segments, more media, more distribution channels, and more price points than ever before—as many as twenty million price points per year for a consumer products company, according to a McKinsey study.⁹ But despite the complex-

ity, what the CMO really wants is to deepen relationships with customers, develop the company's brands, and work with colleagues in other functional areas to grow the business.

Boards of directors now find they must take a more proactive approach to ensuring that their companies survive digital disruption—particularly by overseeing decisions that balance risk and opportunity. They need to make sure the company is building a sustainable position, which, as I'll show, largely depends on building agility and nimbleness into assets and processes. Given the increased pace of competition, they need to find leading or current indicators they can use to assess their company's performance in place of the trailing metrics of traditional financial reporting. Audit committees must ensure that controls are effective despite the increased pace of change, the new risks of the digital world, and the increasing stringency of compliance frameworks.¹⁰

The pattern is that each of these specialist executives must participate outside of their area of specialization by working with colleagues on strategic issues that cut across the enterprise as a whole. The CFO is not just in charge of finance and the CMO is not just in charge of marketing—both are responsible for bringing their functional expertise to bear on *all* of the company's activities and working across silos to accomplish business outcomes. So too for the CIO, who can no longer be responsible solely for running the technology function, but must bring technology expertise to bear on companywide strategy.

The task is harder for CIOs than for the rest of the executive suite, as I'll show in the next chapter. IT was suddenly injected into the enterprise landscape only five or six decades ago and has yet to find its place. As McKinsey reports, "There is little awareness of or agreement on how IT can meaningfully shape a business's future."¹¹ But, the report continues:

... the results suggest one clear element of high-performing IT organizations: active CIO involvement in the business. Where respondents say their CIOs are very or extremely involved in shaping enterprise-wide strategy, they report much higher IT effectiveness than their peers whose CIOs are less involved.¹²

As we move into the digital era, it's IT that can help the CFO, CMO, and the board realize their objectives, supporting them as they move to the strategic role they were meant to play—and indeed *must* play for the digital enterprise to succeed. IT can make the other CXOs superheroes.

As a former CIO, I can tell you that IT needs to be held to higher—but different—standards, and that it will be pleased to step up to them. Engineers are builders—the joy in being a technologist is the joy of creation and of making a difference.

Here's one lesson I learned when I was CIO at Intrax Cultural Exchange. Intrax runs international cultural and educational programs such as work and travel, internships, high school exchange, English schools, and au pair placement. My first large initiative as its CIO was a very successful project to bring our au pair business online. The second project was to do something similar with our high school exchange program; it was a resounding disaster. For that one, IT had received a set of business unit requirements that seemed misguided and contradictory. We were convinced they made little sense from a business perspective. But the business unit was in charge, so in the end we gave in.

We were right, dammit—it was a mess. Employees who had to use it found that it slowed them down. Scrambling to fix it, we made it more like what IT had initially visualized. The CEO called me in and essentially said, “You screwed up.” I protested that we had faithfully implemented the requirements and that it wasn't our fault that those requirements were wrong. To this he replied, “You're missing the point. I trusted you with spending our IT budget and getting good returns. That's not what I got.”

He was absolutely right. A CIO is responsible for investing in technology to achieve business outcomes. It was my failure.

I could have argued that he hadn't set a context in which it was OK for me to disagree with the business unit and reject their requirements. As we saw it then, IT's job was to provide good “customer service.” Taking orders and executing them was what the business unit leader expected. But now as a senior executive, I've come to believe that a CIO has to fight battles when necessary, to use influence and leverage to make sure that the right outcomes are achieved.

Enterprises face pressure to find new ways to grow. Since existing business lines are in danger of being disrupted or lost to competition, companies need to stimulate innovation to protect their markets and forge deeper relationships with customers, finding new opportunities to serve them. KPMG's study of

CEO priorities found that the majority believe growth is more important than finding cost efficiencies, while one-third said their companies weren't taking enough risk to meet their growth objectives.¹³

Geoffrey Moore, author of *Zone to Win: Organizing to Compete in an Age of Disruption*, points out that when a firm enters a new high-growth category, investors price its stock dramatically higher—often to ten times projected revenues or more. Once the category matures, however, valuations stabilize around one- or two-times current revenue. The only way to make the share price move again is to enter another emerging growth category at significant scale.¹⁴ In other words, businesses must be consistently catching the next wave.

In business school I was taught that companies need to develop sustainable competitive advantages (care of the writings of business theorist Pankaj Ghemawat, for example). But sustainable advantages are rare these days. The cloud, the internet, and the globalization of markets have conjoined to reduce barriers to entry. Resource advantages? All companies have access to the same technologies. Locking up a distribution channel is far less sustainable now that new competitors can disintermediate the channel. Firms can build core competencies, as C. K. Prahalad and Gary Hamel have said they must,¹⁵ but who's to say whether their competencies will continue to be of value?

In the digital world, competitive advantage must be constantly renewed. Successful companies continually innovate, harvesting the advantages of each innovation, then moving on to the next when the advantage is competed away. Our economy is largely the one Joseph Schumpeter envisioned in 1942 when he introduced the term “creative destruction.”¹⁶ Growth is driven by innovation, whether it's innovation in products, in building customer relationships, or in improving processes to reduce costs.

The only way to sustain continuous innovation is to reduce the cost and risk of trying new ideas. The good news is that today's IT techniques give companies the agility, nimbleness, and speed they need to do just that. Enterprises using the cloud, along with the set of practices known as DevOps, can deploy IT capabilities to customers and employees hundreds of times a day—rather than once every six months—and can do so reliably, securely, compliantly, and at a high level of quality and usability.

The bad news is while today's technology supports innovation and business agility, the way enterprises use it remains based on mental models from decades ago. The challenge is not in the technology, but in realizing the business value it can deliver. To gain and maintain competitive advantages,

stimulate innovation, delight customers, and react quickly to changing market circumstances, a company must change its way of making technology decisions, overseeing its technology initiatives, budgeting and accounting for its technology . . . and most of all, change its way of interacting with its technology group.

As Stephen Denning says in *The Age of Agile: How Smart Companies are Transforming the Way Work Gets Done*, “Trying to exploit technology and data with the management practices that are still pervasive in many big corporations today is like driving a horse and buggy on the freeway. To prosper in the very different world that is emerging, firms need a radically different kind of management.”¹⁷

The harsh truth is that the C-suite has often not felt comfortable with IT as a business function. How can technologists be held accountable for their work? They’re always late with projects. The systems they create are buggy. Equipment suddenly stops working or is too complicated for employees to use. IT always says no. IT costs are too high; benchmark organizations do IT better and cheaper. IT people overcomplicate everything and speak technical jargon that makes everyone else feel dumb.

The problem is serious. In *Leading Digital: Turning Technology Into Business Transformation*, George Westerman and his coauthors report: “Many executives told us that, given their IT units’ poor performance, they were going to find a different way to conduct their digital transformations. The business executives were going to move forward despite their IT units, not with them.”¹⁸ Particularly disturbing to me is the finding from the CIO Executive Council’s 2015 *Power of Effective IT Communication Survey* that only 3% of business stakeholders consider IT to be a game changer, 11% think of it as a peer, while 58% think of IT only as a cost center or service provider.¹⁹ Although organizations view digital *technology* as a game changer, many apparently don’t think of the stewards of digital technology—the IT folks—as game changers.

In a survey of 800 global business and IT executives, 34% of the business and 31% of the IT respondents characterized their relationship with each

* Note that this is based on a survey of CIOs—that is, CIOs believe that this is the way business leaders think.

other as combative, distrustful, or siloed.²⁰ It is war, in other words, but a war within a single dysfunctional family. And those of us who drive our chariots between the two warring armies feel as bad about it as Arjuna does at the start of the Bhagavad Gita.*

Given this disconnect, it's no surprise that enterprises have thought of bringing a new member into the C-suite: a chief digital officer (CDO). Gartner has proposed that business technology needs to move at two speeds: slow for the legacy, backoffice, and low risk-tolerance systems; fast for customer-facing, innovative systems.²¹ If they're right, then it makes sense to have a CIO who continues to dispiritedly plod along with slow-moving IT and a CDO who's responsible for prancing joyfully about with systems of speed and flexibility.

I don't think much of this two-speed idea. Today's best practices suggest that *all* of the technology in the enterprise should move fast, and that *all* of it should align with the way the company competes in the market and serves customers. Separating IT into CIO and CDO roles is likely to increase the obstacles to nimbleness and innovation I describe in this book.† It would be much better to heal the divide between IT and the remainder of the organization.

But recognizing that organizations have different needs and that some prefer to separate these roles, when I speak of the CIO, please interpret it in the sense of “head of digital technology”—whether that person is a CIO, CDO, or Emperor of Bits and Bytes and First Consul of Computing. In the digital world, an organization will need to learn how best to work with its technologists, whether they're working for a CIO or CDO.

Imagine an enterprise leadership team, who, upon seeing their competitors evolving quickly around them, plops themselves down in a conference room

* In the opening scene of the Bhagavad Gita, the armies of the Pandavas and Kauravas, many of them related to one another, face each other before battle in the Kurukshetra War. Arjuna, a Pandava prince, despairs, seeing his friends, relatives, and teachers on the other side, and asks Krishna for advice.

† Note that my point is only about the C-level position. Digital products should probably have their own product management hierarchy, just like any other product. I would also argue that the C-level leader should act as an advisor/consultant to the C-team on technical opportunities—this also makes more sense if it is a single person.

and proclaims, “We need digital transformation!” They hire a consulting firm or talk to companies that are already enjoying successful transformations. They gather ideas about how leading technology companies move at high velocity. Then they reconvene in the conference room.

“We can’t do that! We’re too bureaucratic. We don’t have the right skills! We have too many compliance constraints! There’s no security if you move so fast! There’s no way to control that kind of IT! Good ideas, but our company is different. It won’t work here until we radically change our culture. But we do *need* to transform!”

Yes, but. . . . Yes, but. . . . Yes, but. . . . This is the leadership team’s impression of bobblehead dolls, heads bobbling in the wake of the fast-moving companies around them.

The bobblehead model is not an effective business strategy.

Now is the time to start transforming, because it’s both low risk and urgent. You might be surprised that I say it’s low risk, but I insist. Enterprises may feel like they should move slowly and cautiously, stepping onto the digital path only after they’ve checked the traffic coming from all directions. But the very point of digital transformation is to *reduce* risk. Digital enterprises set risk-mitigating guardrails* in place, then use their speed as a way to quickly adjust course when new risks appear. And the transformation can be undertaken incrementally, one reversible decision after another. Think: big vision, small execution. This is no time for cautious head-bobbling. You can use the ideas in this book to move quickly and limit your risk.

Maybe it has occurred to you to ask, “What can I do differently to get better results from my technologists?”

Donuts.

It’s worth a try. But maybe that’s just me. My point of view is that of someone who has a background in IT but has seen the divide from the outside as well, having helped senior leaders move their enterprises into the cloud and overcome their cultural, bureaucratic, organizational, and skills barriers. I have been a CEO as well as a CIO, a former software developer (I shall claim to have

* More about risk-mitigating guardrails in Chapter 6: Risk and Opportunity.

been one of the great stylists of the COBOL language—*take that, William Shakespeare!*). I’ve worked in the private sector, the nonprofit sector, and even in government as the CIO of US Citizenship and Immigration Services (USCIS). When it comes to telling hawks from handsaws,* I can train an artificial intelligence “machine learning” program to do it well. I know what as a CIO I wanted my colleagues in other functions to know about IT, and what as a CEO and as an independent advisor, I wanted the CIO to know.

Our goal here is to overcome the IT/business duality so that the enterprise—the business and IT—can enter the digital world, smiling and nodding happily just like bobblehead dolls. Why bobbleheads? To me, there is something endearing about them; they are agreeable and don’t take themselves too seriously. They appear across different cultures—the head-nodding ox called akabeko in Japan; the dancing dolls of Thanjavur, India; Victorian era “nodders”; not to mention today’s bobbleheads that play such an important role in the sacred American ritual of baseball.²² As agreeable as they are, you can’t really tell whether they are bobbling empty-headedly or wisely. A digital transformation, I want to say, means going from a state of agreeable head bobbling to a different state of agreeable head bobbling—one that is filled with wisdom and effective practice. Culture, bureaucracy, risk management, investment oversight—all will continue to be there, just as they always have been.

Throughout this book, you’ll find cycles of creation and destruction—innovation that requires doing away with legacy ideas, bureaucracy and culture that need to be nudged repeatedly in a new direction, Napoleon defeating Russia only to find that Russia has defeated him. But through it all, the bobbleheads bobble. Their smiles will just mean something different at the end of your transformation.

A few themes run throughout:

- For historical reasons, businesses have adopted a model for working with IT that is holding them back.
- This model is deeply connected with a flawed way of thinking about risk and opportunity.
- IT practitioners have long accepted and even reinforced the model, but have recently come to their senses.

* That’s Shakespeare, who had no abilities whatsoever in machine learning. Touché!

- An environment of high risk, uncertainty, and change requires different mental models than does an environment of predictability.
- Software and cloud infrastructure are pliable. Despite some of our preconceptions, they're actually the easiest things in the enterprise to change, and therefore ideal allies in coping with uncertainty.
- Agility is an asset, even if it isn't recognized on the balance sheet.

Part I, "Principles," discusses why the traditional relationship between IT and the rest of the enterprise won't support digital transformation. Part II, "Particulars," breaks down some of the typical concerns the enterprise has about working with its IT organization, and how these can be given a new foundation so as to equip the enterprise for the digital world. "Prescriptions," Part III, is an action plan for moving forward—immediately and with a sense of urgency—into this digital world.

Who this book is for:

If you're a CEO, you are likely to be focused on growth and innovation. If so, you should be thinking hard about how to integrate IT with the rest of your organization. Or perhaps you're thinking about how to nimbly respond to uncertain circumstances while avoiding disruption by startups, competitors, and hackers. I'll show you how to use IT as a strategic component of your organization, yielding results from it that will shape your company's performance.

If you're a CFO, then we need to talk. You're in a difficult position today. You face the challenge of growing your company and finding new sources of value while also safeguarding your company's core business. You have probably recognized that you need to shift your focus from analyzing the past to forecasting and planning for the future. On one hand, you're responsible for control and risk management—you are at the front lines where compliance bureaucracy meets innovation and are the steady hand that manages the company's financial resources and investments. On the other hand, you recognize the need for speed and sense that finance should be a competitive weapon.* IT has always been a problem area for you—one that resists effective controls. I'll

* By the way, Napoleon did too. See Chapter 11: The Leadership Team.

address issues around capitalizing versus expensing IT costs, about new ways of establishing auditable controls, and about selecting investments and managing their risks. How do we measure IT success? What I have to say will, I hope, improve your life.

If you're a non-IT CXO, then this book is about how to let the CIO help you accomplish your objectives. You may be frustrated by the relationship between IT and the rest of your organization, or you might simply have lowered your expectations. But IT is there to accomplish your company's goals. You should be able to rely on its expertise in areas where you've spent less time becoming an expert, but you might need to learn how to help them be more effective. This book should help.

If you're a CIO or IT leader, then you may have begun rolling out Agile development along with Lean and DevOps practices, but have hit organizational impediments in trying to extract their full value. If you read my previous book, *A Seat at the Table*, it may have given you ideas about how to play a more consequential role in your company. Now you need to have a conversation with the rest of the leadership team about it. This book is that conversation.

If you're any other category of business leader, then this book is the missing manual about how to work with IT to be successful. Your performance will be judged largely on what you're able to get from or with IT.

For others in the business and IT community, I hope this book will open up a discussion on the economics and strategic impact of organizational agility through technology. The promise of DevOps was to create a more humane work environment by putting operations and development people on the same team. This book furthers that mission by putting technologists and non-technology business people on the same team so that their heads all bobble together.

PART I

Principles

Copyrighted Material
Excerpt

1

THE BUSINESS AND IT

On the day of cutting off the hair Set and the Company of the Gods fastened my head to my neck, and it became as firm as it was originally. Let nothing happen to shake it off again!

—*The Egyptian Book of the Dead*

A picture held us captive. And we could not get outside it, for it lay in our language and language seemed to repeat it to us inexorably.

—**Ludwig Wittgenstein**, *Philosophical Investigations*

Once upon a time, businesses ran fine without computers. They had pencils; they had paper. A secretarial pool typed documents on typewriters, adding machines added numbers on paper strips, telephones and the postal system provided communications, and the most influential technologies were paper-clips and staplers. It all worked—your company didn't feel like it was missing anything. But suddenly digital technology erupted.

Computers were not like a new kind of stapler or adding machine. The difference—the critical one—was that they could only be managed by specialist knowledge workers. The company had to hire computer programmers, computer operators, and . . . well, other technical folks who required big paychecks. They were experts, but not in your company's business—they were experts in some kind of a science. And soon they became critical to everything your company did. Enterprises suddenly had to find a way to work with these outsiders, to realize business value from their presence. But how?

Imagine we're back in the early days of IT, say 1975 or so. Your company has just bought its first computer: a DigitalWhiz 1000 XPZ (Extreme Processor Zippiness). The DigitalWhiz has its own room, air conditioned to igloo settings. The computer room door has a window that you and other employees can look through as you pass by on your way to the lunchroom. Inside you see refrigerator-sized tape drives jerking in one direction then another, then spinning exuberantly. In the middle of the room is a hunk of machinery you assume is the DigitalWhiz, with a stack of punched cards clack-clacking through its card reader. Paper is jetting out of a fast line printer—the pride of the Computer and MIS (management information systems) Department.

And through that window you can see a few technicians wearing white lab coats, perhaps to keep warm as they thread tapes into the tape drives. They look concerned. One of them you recognize—Gerald, said to be the brainiest of the computer programmers.

You remember the last conversation you had with Gerald. The payroll system had stopped working and employees across the company weren't getting their paychecks. It was urgent, so you found him sleeping under his desk and woke him up. "Gerald, the payroll system is down!" He blinked under his thick glasses, frowned, paused for a moment or two and muttered, "Mmm, interesting." He then moved aside a Star Trek model on his desk, found his M&Ms, and busied himself with them.

"Gerald, it's not *interesting*. The payroll system is down. People aren't getting paid!"

"Ah. Probably the emphatic byte munger I wrote last month. Very interesting. No one has ever munged B++ plackets with Kim-Poppenlooper cyclical nascency before. I was able to integrate a recursive maxicode initializer algorithm into randomized bit-swapper proxy . . ."

"Gerald, this is an emergency."

"Mmm, yes. Would you like me to fix it?"

"Yes! Gerald! Fix! Soon!" You turn and leave him frowning at his desk.

The problem still hasn't been fixed a few hours later when you see Gerald staring deeply at the floor in the hallway. Why, you wonder, had he been spending his time on all those cyclical nascency things when there were bugs in the accounts receivable system, the new inventory system was behind schedule, and the computer department was over budget?

Aside from a few details (I made up the Star Trek model on his desk), this is a true story. As an intern, I worked with "Gerald" during summer college breaks.

Exaggerations aside, there were several lessons in those early IT encounters. It was clear that he and the other “computer people”—the scientists—didn’t care much about the business. What they wanted was to spend their time fooling around with boxes of hardware that looked like B-movie props, writing and rewriting . . . and rewriting . . . Kim-Poppenlooper algorithms. They were odd people and spoke a funny language. They kept spending money to get the latest new equipment, but never delivered what they promised.

They weren’t bad guys (Gerald tried to be helpful), but how could you get them to focus on what the business needed—the reason you were paying for their cyclical nascency in the first place? How could you hold Gerald and his peers accountable when they were the only ones who understood this new IT domain? And every time you talked to him about payroll, Gerald changed the topic to byte-mungers.

So enterprises began hiring IT managers, and later CIOs, who could speak both tech and business. Their goal was to get this new IT *thing* under control; to manage its costs and translate between what had been a fully functioning company before and the new troop of technologists who were suddenly essential.

The stereotypes went both ways. To the IT people, the business was filled with suit-wearing folks who were clueless about technology, were obsessed with organizational politics, were overly demanding, and showed poor judgment in the way they used the software and computers they were given. What they asked for made no sense, and all code had to be protected against their mistakes (“idiot-proofed” was the term they used). It was flabbergasting that the business people couldn’t figure out how to do obvious things, like pressing Ctrl–Alt–Del when the system stopped responding.

A working relationship evolved. The stakeholders specified precisely what they wanted the geeks to do by writing “requirements,” and they set up a “governance” process to rule over IT investments and make sure that IT “stayed aligned” with the business. They had IT prepare Gantt charts and status reports to stay focused on the schedule. With that schedule pressure, IT would no longer be able to waste time on emphatic byte-mungers and dongle fabricators. For its part, IT insisted that the business write down and commit to its requirements so that IT could always prove it had done what it had been asked, no matter how insane.

The relationship was an arms-length, us and them, contractor-like relationship, where something called The Business was the customer of a service provider

called IT. The Business described what it wanted, IT gave a quote, The Business placed its order, and IT reliably failed to deliver on it at the quoted price.

It became natural to think in terms of “The Business and IT” as if they were separate things—the people who tried to get things done to bring shareholders joy, and the technical people who found joy in algorithms and routers. The “and” in “The Business and IT” was more of a wedge between the two—IT people, though they were company employees, weren’t *part* of the business.

Both IT and business folks expressed the hope that someday IT would learn to run like a business, with the CIO as its CEO. Some organizations went as far as setting up a chargeback model, where IT billed each business unit based on the amount of IT it had consumed. IT was expected to provide good customer service to its clientele—the company’s employees. Increasingly, IT resembled an external contractor.

This model let the business feel comfortable that IT investments, expensive as they were, were under control. For IT, it seemed like a way to satisfy the demanding, fickle customers they supposed the business folks to be. Perhaps it was a bit strange that the arms-length service provider group and its customers were actually employees of the same company. Well, not for long; it was an easy step to think about outsourcing IT, since it was already, well . . . outsourced, albeit internally.

The stereotypes solidified and went unquestioned, reinforced by these processes and a bit of confirmation bias. It was easy to catch the geeks doing something that didn’t seem relevant to the problem at hand, or the business folks using their CD drive trays as cupholders. This manner of working “together” became the norm, inscribed in each company’s sacred copy of *The Book of Processes*.

I believe that if you look around today with fresh eyes, these stereotypes no longer hold. Most of the IT geeks I know are very business- or mission-focused, even if they still wear cryptic T-shirts. They love technology—yes—but they are problem solvers more than anything, and love to solve them on behalf of the business. They speak less like geeks and more like post-millennials—because that’s often what they are. The reality is more like that described by Menlo Engineering: “What we have tried to do at Menlo is to emancipate the heart of the engineer, which is to serve others. We engineers exist to produce something that the world will enjoy, something that will delight people.”¹

And the clueless business folks are not clueless. They’ve become accustomed to IT and the consumer technology they use every day; in fact, their

expectations are constantly rising for the technology they're forced to work with. They can make Excel do fancy tricks while the geeks are still pecking with one finger and trying to find the menu option that turns the text red. The business folks, for the most part, have an idea what the cloud is, how many megapixels their camera records, and how to use a computer that doesn't require them to hit Ctrl-Alt-Del. They can tell big data from puny data and sometimes they don't even wear suits.*

Though the stereotypes no longer hold, the processes by which enterprises work with IT are descended from them. But here is a little secret—the stereotypes were never really accurate. I know that when Gerald heard that the payroll system was down, he was immensely concerned. He reacted by going deeply into thought, trying to figure out what had changed that could have caused the problem. And he hadn't been working on his Kim-Poppenlooper algorithm just because it was fun. Kim-Poppenlooper was going to speed up a critical part of the payroll system that was becoming a bottleneck as the company grew, threatening that employees wouldn't get their paychecks on time. In a sense, Gerald knew better than anyone else what was important to the business.

This contractor-like way of working with IT has never been effective. As we all know, projects tend to run behind schedule, IT is generally backlogged and unable to address critical business concerns, systems break down, and employees and customers have to find painful workarounds.

It's more than that. Now that we're in the digital era, technologists—natives of the digital world—should be driving innovation, leading and inspiring the enterprise to make the most of its digital potential. But how can they do so from their arms-length position? How can the non-IT parts of the business change this relationship to get the results they want, all the while managing risk, pleasing the capital markets, accomplishing business outcomes, and seizing growth opportunities? How can they do so in a way that holds IT accountable, provides predictability and transparency, and controls costs? These questions become increasingly critical as the enterprise tries to refine its digital posture.

* JP Morgan now requires all of its employees to take coding classes. "Coding is not for just tech people, it is for anyone who wants to run a competitive company in the 21st century," according to Mary Callahan Erdoes, head of Asset Management.²

What I called in my previous book the contractor-control, or traditional, model for working with IT went something like this: something called “The Business” decides on a new set of IT capabilities it needs. It assembles a requirements document, puts it through an approval process, and signs off on it as a definitive specification. It hands the requirements document to IT and asks for an estimate. After some back and forth, both parties agree on a schedule and budget. IT initiates a project to deliver the capabilities, with agreed-upon milestones for items such as completing a requirements analysis, designing the system, programming the components it will comprise, testing its code, and deploying it to The Business. The Business performs a user acceptance test (UAT) before code is released to make sure its original requirements have been met. And finally, business value happens.

As an IT delivery process, this set of practices is known as the waterfall model, so-called because of how it appears on a Gantt chart. Task follows task and phase follows phase linearly, with milestones in between. Driven by a fixed set of requirements and a project plan, its success is measured by adherence to that plan—“on time” and “on budget” delivery, where “on time” means “as planned before the effort started.”

One seeming advantage of the waterfall is that it holds the technology team accountable. In particular, it’s held accountable for delivering the scope of required capabilities, in accordance with the agreed-upon schedule, and within the agreed-upon budget. Unfortunately, this also forces IT to divide its creative work into separate, ordered phases—each of which must be deemed complete before the next phase starts.

But it turns out that a much more effective way to complete IT work is to perform it iteratively and incrementally, mixing the phases together to get quick feedback from the product users. So what used to seem like a good way to control the IT process turns out to be costly and burdensome to those executing it.

Having observed that the waterfall process wasn’t working, the IT world started to produce books about how to do it better. IT conference sessions discussed how to do a better job of estimating schedules, eliciting requirements, and managing the work so that all phases of projects could be completed according to schedule. Strangely, the result was just more large failures. That is, until around 2001 when a consensus began to form around a new way of thinking about IT projects—the Agile approach, which I’ll describe in a subsequent chapter.

Unfortunately, the old model of interacting with IT isn't likely to lead to the business results you're after. Even if it did what it was intended to do—that is, give you a way to control IT's performance—it would still work against the best interests of your business. This is because the plan-driven waterfall approach depends upon locking in the project scope ahead of time through a requirements document. Since projects are often long-lived, say anywhere from six months to five years, that amounts to deciding in advance what your company will need over that timeframe (actually much longer than that, since the system will continue to be in use for a long afterward).

The more uncertainty there is in the business environment, and the more change that's expected, the less likely it is that the plan accurately captures what will be most important to the company over that period. Rather, today a company should expect that its competitive situation will change, that new technologies will be introduced, government regulations will be amended or rewritten, and new employees will come onboard with fresh ideas. So the company is likely spending a good deal of money to ensure that it'll wind up years behind where it needs to be.

For the plan-driven approach to work, it must avoid *scope creep*—the addition of new requirements after the requirements document is finalized. This is because if the requirements change, then the plan is no longer valid and IT can't be held accountable to deliver on schedule. But as Jeff Patton, an IT thought leader, says in *User Story Mapping*, “Scope doesn't creep—understanding grows.”³ If the company's needs change, or if requirements are discovered to be incorrect—which they will be—then what is truly best for the business? To let the scope creep, or to proceed with the original—and wrong—set of requirements?

The real enemy isn't scope creep, but rather “feature bloat”—that is, unnecessary requirements. It's really feature bloat that results in higher costs and causes the work to take longer than it should. This negative effect is then compounded by the cost of maintaining those extra features coupled with their added complexity. Unnecessary features might even open potential security holes (IT folk say that they increase the *attack surface*).

And yet the waterfall process actually *maximizes* feature bloat by its obsession on eliminating scope creep. Stakeholders are instructed that they must include everything they think they'll need—for the duration of the project through commencement of the next—in the requirements document. Since there is uncertainty about what they'll need, they include every business

improvement they can dream of. A Microsoft study found that only one-third of ideas actually accomplish their intended objective; another one-third have the opposite result, and one-third don't have either effect.⁴ Yet a requirements document includes all three thirds.

The bloat problem has been confirmed by studies showing that more than half of the features in IT systems are rarely or never used.⁵ We've probably all had the reaction, "I didn't know it could do that!" when we see someone use an application's more esoteric features.*

Even entire software applications can go unused. One study showed that across the US and UK, about 28% of the installed software on desktops hadn't been used in the last ninety days, and that the cost of unneeded software amounted to \$7 billion in those two countries.⁶ Since we tend to add features to our systems over time but never remove any, we carry this costly maintenance burden forward year after year.

Companies sometimes admonish requirements writers not to "gold plate" their requests. But in an environment of uncertainty, it's difficult, even impossible, to know in advance which requirements will truly turn out to be valuable. Requirements writers sincerely believe that their requirements are the right ones, at least at the moment in which they are written.

You might think I'm suggesting that companies have been wasting more than half of the money they spend on IT capabilities. You're right—I am suggesting that. Perhaps more to the point for the digital age, imagine how much more quickly you could get products to market if you could avoid all of that feature bloat!

The best way to avoid feature bloat is to start by deploying a minimal product, then adding capabilities only as necessary until the project's objectives have been accomplished. One principle from the *Manifesto for Agile Software Development* says, "Maximize the amount of work not done."⁷ In other words, any work that turns out to be unnecessary—even if it was in the original requirements—should be avoided.

That's to say good technologists add value by *not* doing things. It sounds like cheating, right? Let's say that the project is running behind schedule, but the project team finds a way to get back on plan by not doing some of the

* Did you know that Microsoft Word will let you draw mathematical equations by hand or that it will score your writing based on either the Flesch Reading Ease score or the Flesch-Kincaid grade level score? Maybe you did.

work laid out in the requirements. Should that kind of behavior be rewarded or punished?*

If you think that not meeting some of the requirements is cheating, then you may be possessed by the evil spirit of the IT-as-a-contractor model. The right question to ask is whether the *business objectives* have been achieved, not whether all the requirements have been delivered. Yet when we continue to think of the relationship between the business and IT as a contractual relationship, with IT committed to delivering the agreed-upon scope on a particular schedule, then maximizing the amount of work not done is a breach of contract.

The very idea of assembling business needs into a requirements document has a cost to the enterprise. During the time it takes to accumulate enough needs to declare them a project, assemble the requirements, document them, debate them, and approve them—well, three startups were founded, funded, floundered for a moment or two, found their way, and forever stole your market. Or maybe a competitor's star programmer just built the same IT capability in a few minutes, didn't tell anyone, and snuck the new feature into production.

During all of that prep time the company doesn't see any delivered benefit. Just as there is a time value of money, there is a cost of delay (a metric promoted by Donald Reinertsen in his book *The Principles of Product Development Flow*) in not getting the capabilities into users' hands. One reason we spend a lot of time planning is that we have always thought of IT as expensive. Planning time, on the other hand, seemed more or less free. But implementation costs are lower today and the cost of delay is higher. That's not to say that planning is bad, only that each incremental moment of planning must be balanced against its cost—including the cost of not having the completed work soon enough.

From a risk perspective, the waterfall model dramatically increases the company's exposure compared to the alternatives. It poses the risk that our original requirements and plan won't meet our needs as circumstances change. It raises the risk that some of the requirements are wrong, and since the plan is inflexible,

* In government IT we had a process called Independent Verification and Validation (IV&V), which evaluated the results of each project by going requirement by requirement through the original scope, checking to make sure every one of them had been implemented, thus ensuring that the project team had done its entire job. IV&V, in other words, was making sure that the government had wasted more than half of its money.

IT will be building the wrong thing. And then there is delivery risk; money keeps going into the initiative and results don't come until the end.

Does the waterfall, in fact, give the enterprise predictability and control over its IT investments? Is the pope Zoroastrian? Does a bear shave in the woods?

The waterfall tries to make sure the project stays on schedule by conducting status meetings. IT reports on its progress relative to the Gantt chart with—you know—those little red, yellow, or green circles to indicate whether it's on track. I've seen too many of these; they're all the same. The current work is always somewhat behind schedule, but yet the remainder of the schedule is still on track. The delays are temporary and later work will make up for them by taking less time than planned.

Suddenly, one week the little yellow balls turn red! The project team, grilled on the cause of the "delay," draws some ill-founded conclusions about why it's behind schedule, promises to fix the underlying problems, and guarantees that it will get back on schedule.

Well, of course they say that. They're already committed to the original schedule, right? Note that the best, most logical explanation for what is happening is that the original estimates were too low. And yet the team promises, against all evidence, that those for the next phase are too high!

This is a consistent pattern in cases where uncertainty is manifest. The original plan was known to be subject to uncertainty (a true estimate really would include an attached confidence interval, and a large one at that), but when the uncertainty actually manifests itself, "reasons" are found and blamed. The truth is that those reasons are statistical noise—we already knew there would be variances.

Here's a game. I show you ten boxes, each containing a bobblehead. I tell you that seven of them contain bobbleheads of Blaise Pascal and three contain nodding images of Reverend Thomas Bayes. I point to a box and ask you to guess who is inside. You, very intelligently, guess Blaise Pascal, but when I open the box, there—grinning and bobbling—is Reverend Bayes. Do you then blame yourself for the mistake, try to figure out the "reason" it was Bayes rather than Pascal, and vow not to make the same mistake again?

Many businesses believe they need *predictability* in the delivery of IT capabilities—defined as conformance to the original project schedule. You

might want to think carefully about whether or when you do need it. How important is predictability to you, relative to business agility and effectiveness? And to what extent can it be achieved in a complex and uncertain environment?

I can imagine some cases where predictability is very important. For example, the company might be preparing an IT system that has to be ready by a certain date, as was the case with the Healthcare.gov fiasco that was timed to coincide with a political initiative. Or it may be a situation where other company plans—perhaps marketing activities—must be coordinated with the launch of an IT system.

Unfortunately, the waterfall doesn't actually offer predictability, since complexity and uncertainty often don't cooperate with any plan milestones. And as we've seen, project teams are incentivized—required, really—to deny the reality of schedule changes until it's too late to maintain the fiction of being on schedule.

If predictability is important because other activities need to be coordinated with the completion date, then it's best to have the team constantly re-estimate and re-baseline the schedule based on their progress. How often should they be allowed to do this? As often as possible. The more often they re-baseline, the better predictability you'll have. This isn't predictability in the sense of knowing before the project starts how long it'll take; rather, it's predictability that acknowledges the reality of change and uncertainty.

But predictability isn't really the issue here, is it? A demand for it is often a way to try to enforce control over a project, to insist that the project team deliver on its "commitments." It's a way to hold IT accountable, not a way to gain predictability. This is my point—many of us have become trapped in a mental model that has become so ingrained that we barely notice it is there.

It sounds strange to say, but what you want from IT is not delivery *on schedule*. You want it *as soon as possible*. This is very different. If it can't be delivered as scheduled, then you don't really want it to be. But if it can be delivered ahead of schedule, then that's what you want. You really want urgency. You want delivery ASAP. When you think about it, ASAP delivery means the same thing as delivering with the shortest possible lead time, which is the goal of Lean manufacturing. You can take out your Lean playbook and ask, "What are the

steps in the delivery process? Which of them can be shortened? Where can we eliminate waste in the delivery process?”

That—precisely—is what you want to hold IT accountable for: the leanest, most waste-free, shortest lead-time delivery of needed capabilities. And if it turns out that some of the waste in that delivery is outside of IT, or within the interactions between IT and the rest of the business, you want to eliminate that as well. Spoiler alert: some of it is.



The classic view of IT is not just about enforcing control through pre-planning and milestone adherence: as we’ve seen, the contractor-control model also encourages IT to treat the other business employees as its customers, further distancing IT from non-IT. George Westerman and his coauthors describe and dismiss the customer service model in *Leading Digital: Turning Technology Into Business Transformation*:

In the long-distant past, we were taught that IT was the keeper of technology and that IT leaders were service-providers to the rest of the business. Their job was to stay aligned with business strategy, taking orders from the business and delivering new systems. If they kept the systems running and delivered new projects on time, then all was good. That time is over, and has been for many years.⁸

That time is over, because merely keeping systems running and delivering projects on time is both too much and too little to ask of IT. Too much, because there is too much uncertainty in the IT world. Too little, because even if IT could deliver these things, it wouldn’t necessarily be delivering business value. It could easily be delivering the wrong things on time, or overspending on keeping systems running. In the digital world, you don’t want IT to be a service provider—you want it to deliver business outcomes.

When IT has functioned as an order-taker, it has predictably wound up with too many orders to manage. To paraphrase blogger Pascal van Cauwenberghe, IT winds up with a “vomit of requirements” that it must then clean up.⁹ Too much demand means long wait times to get important work done, and because IT is jerked from one demand to another, it cannot formulate a meaningful IT strategy or plan an overall architecture for the company’s systems.

The contractor-control model also disincentivizes some of the behaviors you seek from IT. To keep the company's infrastructure secure, you need IT to be an enforcer, sometimes making its customers unhappy. Depending on budget decisions you make, IT may need to say no to tasks or devices its customers want. If employees have misunderstandings about how to use technology, IT should set them straight. And when requests are impractical or against the best interests of the business, IT should refuse. This is customer service without the smile. In Westerman and Hunter's words:

Saying that "the business is IT's customer, and the customer is always right" seems like a good idea. . . . But over the long term, this value trap sets up the IT unit for failure, because customers are often wrong (especially about matters in which they are not expert), and calling colleagues "customers" puts a wedge between IT and the rest of the business.¹⁰

Customers are the people outside the enterprise who pay for its products. IT people and non-IT people within the enterprise are *colleagues*.

With no better model for what success looked like, IT leaders often took the position that they would "run IT like a business." They would look for efficiencies and benchmark IT against outside organizations. They would institute a chargeback model, where lines of business were charged based on the amount of IT service they consumed. Geoffrey Moore, usually a more insightful writer on IT subjects, explains this model's presumed benefits by saying that "it puts the service-providing organization on notice that program work is not an entitlement but rather must be earned, potentially in competition with an external supplier."¹¹

This places the IT organization—employees of the company, note—on notice that they could be replaced by an outside supplier. Honestly, doesn't that sound like an unpleasant way to treat employees? How have we come to think of this as normal?

The idea is a poor one for other reasons. While the chargebacks are supposed to reduce demand, they inevitably fail to result in an equilibrium where supply matches demand. Of course, there is a way to achieve such a balance, and that is to let IT's prices go up until the market-clearing price is reached. It

would be a price unrelated to actual costs, and the IT group would show a profit at the expense of the other budget holders . . . who would not be happy.

If IT really were an independent business, it would be able to scale as needed to meet demand. It would be able to hire freely and set salaries optimally. It would be able to choose its customers, only taking on new ones when it had available capacity. No, IT is not a business, and we would not want it to be.

It would be curious if an internal IT department lost a benchmarking competition against an outside organization. An external contractor must earn a profit. It has transactional costs that it must pass on: the costs of negotiating a contract, legal fees, and administrative fees, along with sales and marketing costs. It has costs in coming up to speed on its customer's business and its organizational dynamics. When changes must be made to requirements, not only does the contractor add fees, but the company suffers administrative costs in communicating those changes and authorizing the extra charges.

Nor is the argument convincing that the outside organization is better or more cost-efficient; IT most likely uses similar delivery practices, and it can pull from the same labor markets to hire similar people. Yes, economies of scale can give the outside provider an advantage, but only for services that actually have economies of scale. Cloud infrastructure, for example, can be obtained at a higher quality level and lower price because of scale. But the usual functions of IT as a service provider—basically the provision of human effort—do not lend themselves to such economies.

In an apples-to-apples comparison, there is simply no way an outside service provider can provide *the same services* at a lower cost. The IT department can only come up short if the benchmark is comparing rambutans to durians. The internal IT group may have a better sense of what will actually be required, while the external contractor will wind up adding to their charges when the enterprise later submits change requests. The external contractor might compromise quality or not pay as much attention to making the system easily maintainable or secure. Or maybe your development work is a loss-leader for them and they plan to profit through subsequent system maintenance work.

Treating IT as a contractor can unnecessarily impose many of these costs on the enterprise. CIOs are sometimes told they must market their services across the enterprise and demonstrate that they're adding value. This is waste—IT should focus all of its resources on actually adding business value. Back-and-forth negotiations between IT and business stakeholders are also a cost that the enterprise shouldn't have to bear. And chargebacks to business

units—while they might be valuable for cost accounting—add administrative costs and frustration. Avoiding these should be an advantage to keeping IT inhouse!

All of this is beside the point in the digital world. Treating IT as an external service provider sacrifices its biggest advantage as an internal entity: the fact that it can have closer contact, more touch points, and more involvement in running the business itself. An internal IT department cares about the company's mission, absorbs its culture, and participates in defining what the company is.



To compete in a digital world, you must engage the IT department in determining and fulfilling business objectives. IT cannot be accountable merely for producing IT products as “required,” but instead must be given, and must assume, accountability for business outcomes. Your IT technologists are your colleagues and allies in times of digital disruption.

6

RISK AND OPPORTUNITY

Tragedies and comedies are written with the same letters.

—**Democritus** cited by Aristotle, *On Generation and Corruption*

Some white-haired old man, with many rings on his fingers, will come along and shake his head, and say to me, “Listen to me, child, yes, one ought to practice philosophy, but one should also keep one’s head.” This is sheer stupidity.

—**Epictetus**, *Discourses*

Risk is the possible *negative* consequences of the uncertain future, while opportunity is the possible *positive* consequences of the uncertain future. Agility is the organizational characteristic that determines whether the uncertain future becomes one or the other—or simply a benign passage of time.

For those who worry about risk, there is much to worry about. Startups arise suddenly and disrupt industries. Customers are fickle and change their tastes quickly, sometimes within the few moments it takes to read a tweet. Countries Brexit or dismiss treaties they have previously signed. Regulations are put in place and then rolled back. And technology changes in nanoseconds—when you pause for a bio-break your programmers may begin studying a new programming language you’ve never heard of; when you return they’ve already rewritten your IT systems. Both the business and technology environments are uncertain and unstable—risky, many would say.

Uncertainty, however, doesn’t always lead to negative consequences. On the contrary, it also opens up opportunities. When an unforeseen event occurs,

what determines whether it's an opportunity or a hazard? Yup—agility. “It is possible to prepare for unknown risks,” reads a Boston Consulting Group (BCG) blog post,¹ to which I would add that it's possible to prepare for unknown opportunities as well. The post goes on: “[Our] research has demonstrated that highly adaptive companies outperform less adaptive companies in periods of economic turmoil.”²

Let's say you spend \$10 million to build a factory to produce bobblehead dolls of Sir Isaac Newton, everyone's favorite scientist. Suddenly Albert Einstein announces some theory of relativity and the market for Isaac Newton bobbleheads immediately dries up. Has your \$10 million investment been sucked into the proverbial black hole?

Yes, if your factory can only produce Queen Anne-era Cantabrigian scientist dolls. But if it's agile enough to quickly retool to produce Einstein dolls, then your investment becomes worth even more: you'll be first to enter the relativity market and have a first-accelerator advantage. In other words, designing the factory to be agile from the start lowers your risk—you are able to deal with changes that take place in the uncertain future.

Given an uncertain future, anything that increases your cost of change also increases your risk. Anything that decreases the former also decreases the latter. Reducing the cost of change is the definition of agility. To put it bluntly: risk is lack of agility.

Another equation: “Risk management is strategy,” BCG says, “and strategy is risk management.”³ In some cases, they can barely be distinguished. Let's say you choose to cross over from the bobbling head market to that of those cats waving their paws up and down—the ones you see in many Japanese restaurants. Is your decision to build a bobblepaw* cat a strategic product extension? Or is it mitigating the risk of the bobblehead market drying up? Or of competitors moving more quickly into the bobblepaw market? There really is no distinction. Risk management is just a different way of looking at strategy.

There is risk in both stasis and change; risk in competitive actions and in passively responding to competitors' actions. An analysis of the one hundred companies having the largest stock price drops from 1995–2004 showed that

* They're called *maneki neko*, not bobblepaws. Poetic license.

only thirty-seven were hurt by financial risks, while sixty-six were hurt by strategic risks—including competitors' actions, for example.⁴

An enterprise's goal is not to eliminate risk, but to use it for competitive advantage. If the enterprise could, say, put in place robust automated controls to reduce risk—seizing opportunities more aggressively as a result—then it could also seize market share. If it could eliminate controls that slow it down without effectively mitigating risk, it could achieve a speed and cost advantage. And if it could assess risk better than its competitors, making better bets as a result, it would wind up ahead.

The cloud, DevOps, digital transformation—all of these come with techniques for increasing business agility, thereby reducing risk. Yet enterprises sometimes hesitate to adopt them because their leaders perceive risk in the unknown. It's an interesting kind of risk: one where we don't really know what the risk is. Our fear is of a second order: the risk that there might be a risk.

In another USCIS incident, a number of us met to discuss the severe problems we were having with the performance of a large contractor. At one point, someone suggested we start a new request for proposal (RFP) to replace the contractor. "Too risky," said one of the more senior executives. "We don't know what kind of a contractor we'll wind up with or how good they'll be."

I've heard many variations of this line of thought. In this case, we had a contractor who had a 100% chance of performing poorly (since it was already doing so), yet the perceived risk of working with an unknown one somehow seemed to be higher. It's equally strange that some organizations are afraid of Agile and DevOps practices, as these were invented as a way to reduce risk. Ditto for the cloud, which eliminates the risks of managing onsite infrastructure.

I've had conversations with security experts in the government and at large commercial enterprises where someone will ask, "Is the cloud secure enough?" It is, but this is the wrong question. What they should be asking is, "Where will my security posture be better—in the cloud or in my on-site datacenter?" The framing of this question reveals a lot about fear of the new. In my role at DHS I found the decision easy, by the way. The cloud clearly enabled us to build a much more robust security architecture.

Ask any information security specialist if they're happy with their security posture in their datacenter:

“No way. Too many people have privileged access;* we have too many insecure legacy platforms; we don’t patch often enough; our firewall rules are too complex; production systems aren’t reviewed often enough . . .” and on and on.

“How about moving to the cloud, then?”

“Well, that would be risky . . .”

There’s a pattern here: we tend to attach too much weight to the risk of the new and too little weight to that of the status quo.

In fact, this is an instance of a common cognitive bias, described in a 1988 article by William Samuelson and Richard Zeckhauser, “Status Quo Bias in Decision Making.”⁵ The authors’ experiments showed that people disproportionately decide to stick with the status quo when presented with alternatives. And in a 2016 *Psychology Today* blog, Rob Henderson wrote, “Status quo bias is a cognitive bias that explains our preference for familiarity. Many of us tend to resist change and prefer the current state of affairs.”⁶

Status quo bias was further explored by Daniel Kahneman, Jack L. Knetsch, and Richard H. Thaler in their paper, “Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias.”⁷ They relate status quo bias to a phenomenon called the endowment effect—the tendency of people to give a higher weighting to things they already have when making decisions.

What at first seems like fear of the new is perhaps better thought of as an emotional preference for what we already have. The effect is stronger the more choices we’re confronted with (think of all the options available in the cloud!), and is stronger the longer we’ve held the object we may be giving up. It reminds me of those groovy old COBOL mainframe systems that have been around since hippies occupied Golden Gate Park during the Summer of Love. Hard to give up, right?

There are all sorts of risks in today’s business technology environment—things for managers and leaders to worry about. There is risk that a large IT investment won’t return its intended business benefits. There is risk that a dis-

* That is, they are given special access to systems so that they can perform administrative tasks that are not common. These people are usually IT employees, or perhaps business people who manage the access of other users.

ruptive startup will shake up the industry. Or that a hacker will steal sensitive customer data. What about a competitor thinking of a brilliant idea first? Then there's risk that costs will spiral out of control. Or that a new technology will make the current infrastructure obsolete.

I could go on and on. There are so many risks it's a wonder that an enterprise can do anything at all. But that's exactly the point—the biggest risk is not change, but stasis. Or as the Marine Corps doctrine dryly puts it, “Risk is equally common to action and inaction.”⁸ Unless you're sure that your enterprise is already prepared to meet all of the aforementioned risks, the status quo is a terrible place for you to be, and the risk of the new should seem negligible compared to the urgency of change.

To be very clear, I'm not suggesting that companies change their risk tolerance—if anything, I think they should be *more* risk averse. There has been persistent confusion in the literature of digital transformation, with writers suggesting that companies need to become more comfortable with taking risk, that they should consider failure to be a good thing . . . just another learning opportunity. I'm talking about comments such as, “We need to encourage risk taking and quirkiness” in Jim Highsmith's book *Adaptive Leadership*.⁹ Yes, I'm all for quirkiness. But do we really want to encourage risk-taking? I think Highsmith doesn't really mean what he means. He is perhaps thinking of Xerxes, poised to invade Greece at the Hellespont:

If you were to take account of everything . . . , you would never do anything. It is better to have a brave heart and endure one half of the terrors we dread than to [calculate] all of the terrors and suffer nothing at all . . . Big things are won by big dangers.”¹⁰

But business leaders have a fiduciary duty *not* to take on big dangers, though there is a lot to be said for proceeding with confidence once a decision has been made (it didn't work out too well for Xerxes, by the way). The miscommunication, I think, comes from using old terminology and mental models to describe the new ideas. When a digital transformation writer says that it's important to “fail fast” and encourage failure, what they mean is that it's a good thing to try experiments and abandon them if the results aren't the desired ones, or to change direction frequently if change is warranted.

Experimentation and changes of direction, however, are tactics to *reduce* risk. Instead of committing to an investment such as building a product,

choosing a technology, or launching a change to a digital service (all risky), a more Agile approach is to make a smaller commitment to an experiment first, then gauge whether the larger investment will be effective. It's a case of buying information to reduce risk. If it turns out that the larger investment isn't a good idea, then the experiment was not a failure but a success; it gave us critical information that let us avoid making a bad investment.

So please do not fail. And if not failing works out well as a strategy for you, please tell people you read about it here.

The experimental approach can also reduce the risk of fixating too early on one idea when better ideas might be available. Experiments can be rigged to try a few different innovative solutions, then compare the results. If one idea turns out to be better than the others, then testing alternatives wasn't a failure—it was a success at inexpensively buying information to reduce risk and make a good decision. As Yogi Berra said when giving directions to his house: “When you come to a fork in the road, take it.” Experimentation encourages innovation by allowing ideas to be tried out, then vetted by finding Yogi's house.

Experimentation is a powerful way to avoid analysis paralysis. Since there is a cost of delay, it's often better to proceed on a small scale with a reversible decision, then quickly pivot if it doesn't work out. The cost of the effort spent going down the unsuccessful path is likely to be less than the cost of a long analysis, and the conclusion is more certain. Proceeding quickly with reversible decisions *reduces* the risk of being late to market. That's success, not failure.

You can also think of the experimental approach as a strategy of doubling down on winners. Experimentation allows us to place small bets on a portfolio of possible winners. When we find the odds on one particular idea increase, then we raise that bet.

Doubling down, by the way, is much more than a cliché—it's the actual strategy in blackjack. If you're dealt cards that add up to ten or eleven, then the only correct strategy is to double your bet. This is hard for many people to get used to—they view doubling down as optional. But in fact, over the long run, if you want your odds to come close to being even with the house, you *must* take advantage of the opportunity to double down when the cards look good. In other words, doubling down on likely winners in your “portfolio” of blackjack hands is a way of playing your cards to *reduce* risk.

The experimental, tentative decision technique is neither about failing fast nor taking more risks. As Highsmith says, “Traditional teams attempt

to drive out uncertainty by planning and analysis. Agile teams tend to drive out uncertainty by developing working software in small increments and then adjusting.”¹¹ Why then does he advise taking risks?

In a predictable world, perhaps it’s right to consider anything new and unknown to be risky. In a world of uncertainty and change, however, anything old and known is risky. It’s risky as well to blindly follow a plan, because we know that the world is changing and our plan is based on what we knew yesterday. This is difficult to accept, because plans have always been our way of reducing risk. Today, though, the unexpected remains unexpected even for careful planners.

The more risk-averse you are, the more excited you should be about digital transformation, even if it’s frightening because it’s new. Frightening isn’t the same as risky.

It turns out that people are pretty terrible at assessing probabilities and risk. In my last book, *A Seat at the Table*, I cited several examples to make this point—examples that I love because even knowing the right answer I still can’t convince myself it is right.

The first example had to do with the TV game show *Let’s Make a Deal*, in which the contestant is asked to choose one of three doors. Behind one, the contestant is told, is a car the contestant will win if he or she guesses correctly. Once the contestant guesses, the host opens one of the other doors to show that there’s no car behind it. He then points out that two closed doors remain—the one the contestant chose and the third one—and asks if the contestant wants to switch doors. Since there are now only two doors, each seems to have a 50% chance of hiding the car, so it doesn’t matter, right? But the correct strategy is *always* for the contestant to switch doors—doing so doubles his or her chances of winning.¹² Our intuition hesitates to accept that, even when it’s carefully explained.

Another example that I cited looks at probability in the world of disease. Let’s say you’re tested for a rare disease that occurs in one out of every one thousand people. The doctor informs you that the test is 99% accurate: that is, if you have the disease, the test will be positive 99% of the time, and if you don’t have it, the test will be negative 99% of the time. Unfortunately, the test comes back positive. How worried should you be? Not very, it turns out. With

these parameters, the chance that you actually have the disease turns out to be only about 9%.¹³

We have a strong tendency to identify risk in all the wrong places and miscalculate the likelihood of outcomes. Traditional project management, for example, leads us to believe that projects have cost, schedule, and scope risks. A project manager following Project Management Body of Knowledge (PMBOK®) best practices will keep a risk register and make plans for mitigating those risks.

But these delivery risks are not the right risks to be concerned about. The real risks are (1) the risk of not accomplishing our business objectives, and (2) the risk of not accomplishing them in the quickest, most cost-effective way. To those I'll add (3) the risk of unintentionally exceeding the budget. Cost, schedule, and scope are at best only proxies for these real concerns.

(1) The Risk of Not Accomplishing our Business Objectives

If your goal is to increase the number of cases each employee can process in a day, say from seventy to one hundred, and you're willing to spend a million livres to achieve that objective, then the risk is that you'll spend your million livres and not achieve the objective.

If you take that objective and translate it into a set of requirements for technologists to implement, then you've added another risk—the risk that the requirements you've chosen won't actually accomplish the objective. Yet that's exactly how the traditional model proceeds; it does little to mitigate this risk, since you don't find out whether your requirements have succeeded until the project is over and you've spent all your money.

You can reduce the risk by taking the Lean startup approach and treating the so-called requirement as a hypothesis: "I believe that if we implement these features, then the result will be to increase cases from seventy per day to one hundred." Then you'll test the hypothesis, often with only a fraction of the total spend of the project. Based on the results of your test you can either modify the requirement or abandon it altogether. *That* is risk mitigation.

You can further mitigate this risk by using DevOps to quickly release one capability at a time to users. You can then see the objectives being accomplished as development proceeds; in the prior example, you can see the number of cases continually increase from seventy. With every released feature, you're decreasing the risk of not accomplishing your objective. What better risk mitigation could you ask for?

(2) The Risk of Not Accomplishing Them in the Quickest, Most Cost-Effective Way

With the old waterfall approach, it was almost certain that you were not accomplishing objectives in the quickest, most cost-effective way. The waterfall, remember, actually increased risk by encouraging feature bloat. It required costly upfront time to prepare the requirements and to plan the project before delivering any value. And it increased costs because testing came at the end, when it was the most time-consuming to fix problems (see Figure 4: Cost to Fix a Defect Versus Feedback Time).

Fortunately, your toolkit now includes a better way to manage this risk as well. With an Agile approach you can begin delivering value right away, evolving the plan as you proceed. You reduce cycle times and eliminate waste by looking at the entire value stream, both inside and outside of IT. All of these reduce the risk that you won't accomplish your objective in the quickest, most cost-effective way.

(3) The Risk of Inadvertently Exceeding Budget

In the waterfall approach, we treated scope as fixed (“required”) and continued a project until either the scope had been completed or the project had been terminated as a failure. Because the project had to continue until the scope was complete, there was a high risk of exceeding budget. That was precisely what happened on many projects.

When you use the Agile approach, you get results constantly throughout the process, prioritized by their effect on the desired outcome. As a result, if the budget runs out you can simply stop the project—it has already delivered most of its value. Or you could decide to increase the budget and produce more impacts. You have the choice.

Disaster! An IT system has stopped working! Bobbleheads are bobbling in vain—they can't find their way to the people who want to buy them. The warehouse is filling up with rows and rows of grinning, nodding superheroes and politicians. The CEO, caught unprepared before his first nespresso of the day, angrily texts the CIO demanding justice for the homeless bobbleheads. Two hours later they still aren't moving—horizontally, that is. What a terrible reflection on the quality of the IT systems!

Or is it? Availability costs money and fluctuates uncertainly. At some point the company probably made decisions about how much to invest in system availability. Perhaps it was willing to accept three nines of availability (99.9%) rather than spending the incremental dollars it would have cost to achieve four nines. Three nines means that the system is expected to be unavailable about nine hours a year. If the bobbleheads have been bobbling futilely for two hours now, is there an availability problem?

We simply don't know. Over the long term we're expecting about nine hours of outages, so this two-hour period may only be a part of that total duration. If today is January 25th and the system has been offline for two hours this early in the year, does that mean that it'll be down for twenty-four hours over the year (clearly unacceptable)? And does availability of three nines mean it will *predictably* be down for nine hours during the year? Wouldn't it be strange if we hit that number exactly? Shouldn't we expect some variation above and below the third nine? In that case, what if today's outage lasted eleven hours? Is that OK?

The answer to all of these questions is that, at the moment, this outage is statistical noise.* If the company reasoned correctly when it set a target of three nines, then everyone should stay calm. It merely confirms that the company saved money by not making its system more available than planned.

"Something must be done!" the CEO texts the CIO all in caps. A CIO who understands probabilities should relax and finish their breakfast.

Many risks are managed by putting controls in place. This is especially important when the organization has to fulfill compliance requirements such as SOX, HIPAA, PCI, and FISMA. Sooner or later the auditors will come around and try to verify the controls. The organization that is practicing good digital hygiene will have a good story to share with them.

In the digital world, controls are automated to the greatest extent possible. Automated controls are more reliable than manual ones and are more efficient to apply. They leave an audit trail, and we can take advantage of cheap storage to retain vast quantities of auditing information for as long as

* I'm over-simplifying. We can actually calculate a revised probability of exceeding our target given the new information that it has been down.

necessary. Because automated controls are applied continuously (rather than just periodically), technologists can work quickly, knowing that they're always compliant. In the DevOps world, controls, therefore, speed up work rather than slow it down.

Security controls, for example, can constantly test a system's security as it's being built. If a technologist makes a mistake that creates a vulnerability, the tests spot it right away and provide feedback with which the technologist can immediately fix the problem (and learn from it). When the system is finished, there is no need for a long period of security testing and validation; it has already passed its tests. As a result, lead times are shortened.

Automated controls can be substituted for manual ones in many more areas beyond information security. Cost controls can be put in place in the cloud to limit spending and ensure that infrastructure is tagged with accounting cost categories. Privacy controls can be used to restrict access to data. Approval workflows can be set up where necessary—although often even the approvals can be automated. In the government we had to comply with section 508 of the Rehabilitation Act, making sure that all systems were accessible for users with disabilities. We found that most checks for accessibility could also be automated.

Let me illustrate how controls can speed up delivery with an example that was at one point controversial. It involves the separation of duties between software developers and system operators (those who can make changes to production systems), previously considered to be an essential control. A developer who created code had to give it to an operations specialist, who would then validate that it was production-ready. This meant checking that testing had been completed, that users were ready to receive the new features, and that the deployment process could be completed with minimal risk.

In a mature DevOps process, however, deployments are automated. Many organizations simply permit the developer to “press the deploy button” without having to hand the code off to someone else. At first this seemed to violate the principle of separation of duties, but many auditors have realized that the automated DevOps process actually has better safeguards.

First, the automated process can ensure that the code has passed all of its tests before it's deployed. Since developers can deploy quickly and freely, they're able to deploy smaller bits of code more often. This reduces the risk of defects and the impact of changes on users. Every change is tracked in the version control system and can be audited. The testing process—which again

must pass before the code is deployed—includes security and compliance tests. Many delivery teams also require code, after it's written, to be immediately reviewed by a peer developer. Better equipped to spot problems than an ops specialist, this person then provides feedback through an automated system (again, so it can be audited).

The automated controls, in this case, make for a much faster process and increase control. We can have our donuts and eat them too.*

Entering the digital world doesn't really mean taking more risks, though some people speak of it that way. It's rather a matter of correctly understanding what the risks are, then relying on fast feedback and hard data to mitigate them in a different way than we have in the past. Properly understood, fear of the new is not a risk at all.

The job of the change agent becomes much easier when you stop suggesting that your enterprise take more risks. Instead, you can carefully identify the real risks and craft an Agile strategy to manage them.

* If you are confused by the proverb, you are not alone. The original sense was as in the 1546 version: "Wolde you bothe eate your cake, and have your cake?" In other words, you can't continue to have it after you eat it. In Iceland they say, "You cannot both blow and have flour in your mouth." I don't understand that either.