

The background of the cover is a dark blue gradient. It features a complex, glowing network of white and light blue lines and dots, resembling a digital or neural network. Some dots are labeled with the numbers '0' and '1', representing binary code. The overall aesthetic is high-tech and digital.

DIGITALLY REMASTERED

Building Software into
Your Business DNA

Otto Berkes

Digitally Remastered: Building Software into Your Business DNA

by Otto Berkes

Afterword by R “Ray” Wang

Founder & Chairman at Constellation Research, Inc.

WILEY

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor:

Paul Levesque

Business Development Manager:

Felicity Whyte

Editorial Manager:

Rev Mengle

Cover photo:

John McNeil Studio

Production Editor:

Vasanth Koilraj

Digitally Remastered: Building Software into Your Business DNA

Published by

John Wiley & Sons, Inc.

111 River St.

Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2016 by CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The statements and opinions expressed in this book are those of the author and are not necessarily those of CA, Inc. ("CA").

No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-119-27153-6

ISBN-13 (electronic): 978-1-119-27383-7; 978-1-119-30156-1

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Brief Table of Contents

Introduction 1

1 Remastering Your Business for a Digital World 3

2 A Blueprint for Your Modern Software Factory 11

3 Building Your Software Factory 21

4 The Agile Journey 31

5 APIs — Your Digital Nervous System 41

6 Integrating Security to Fuel Your Transformation 53

7 Measuring for Insights. 61

8 Innovating to Deliver Customer Value 67

Conclusion. 75

Afterword 77

Endnotes. 79

Table of Contents

Introduction	1
1 Remastering Your Business for a Digital World	3
What's Different Now?	4
Redefining Technology's Role in Business	4
Every Business Is in the Software and Service Business	5
Customer Engagement through Software	6
Building a Modern Business with Software at the Center	7
Welcome to Your New Operating Environment	8
What's Next	9
2 A Blueprint for Your Modern Software Factory	11
Optimizing for Continuous Development and Delivery	12
Core Operating Principles of Your Modern Software Factory	13
Two Software Cycles: Experience and Service	13
What <i>Should</i> Your Software Factory Look Like?	16
Getting Started	18
3 Building Your Software Factory	21
Delivering Software Continuously by Optimizing 'Below the Line' Activities	21
Building Your Factory Team	23
Common Obstacles to a Smoothly Operating Factory	25
Getting Started	27

4	The Agile Journey	31
	The Road to Agile	33
	Scaling Agile	34
	Roadblocks to Agile Adoption	35
	Getting Started	36
5	APIs — Your Digital Nervous System	41
	Building a Connected Business Using APIs	42
	The API Advantage	44
	API Challenges	47
	Getting Started	49
6	Integrating Security to Fuel Your Transformation . . .	53
	The Security Imperative	54
	Identity Is the Perimeter	56
	Getting Started	57
7	Measuring for Insights.	61
	Exploiting the Data You Have.	61
	Common Challenges with Metrics and Measurement	64
	Getting Started	65
8	Innovating to Deliver Customer Value	67
	Finding the Sweet Spot	68
	Getting Started	72

Conclusion.	75
Afterword	77
Endnotes.	79

Author's Acknowledgments

When I was approached with the idea of writing a book on digital transformation, it seemed like a straightforward process. Little did I know that countless weekends and other previously free time would be consumed by a seemingly endless cycle of discussing, writing, editing, and reviewing the words and the ideas they represent on the pages to follow.

I didn't realize just how much of a team effort this book would require, or that the process of writing it would utilize key concepts such as cross-functional collaboration and rapid iteration that are important topics in the book itself.

Justin Vaughan-Brown captured the initial concept and rough outline, Denise Dubie, Jason Meserve, Pete Waterhouse, Kieran Taylor and Rick Langsford contributed content and expertise for the first drafts, and Ryan Martens, Ronica Roth, Bill Talbot, Sumner Blount, Tyson Whitten, Huw Price, and Chris Rowett provided interviews and feedback on early versions of the narrative. Robyn Rawlings, Jason Meserve, Ed Cone, Adrianna Gregory, and the Thought Leadership team at Oxford Economics drafted the customer case studies, and the team at John McNeil Studios — Kim Le Liboux and Edgardo Sanchez — created the graphics and cover design.

R “Ray” Wang with Constellation Research, Guljeet Nagpaul with Cognizant, Munawar Lakdawala with PwC, and Kalyan Kumar with HCL Technologies wrote perspectives that are included throughout the book. A special thanks to CA's visionary customers for allowing us to share their digital transformation stories — Advisory Board Company, DHL, Lockheed Martin, L'Oreal, Orange, PayPal and Williams Sonoma.

My CTO review board — consisting of George Watt, Richard Philyaw, Craig Vosburgh, Mordecai Rosen, Joav Bally, Debra Danielson, and Paul Dulany — added thoughtful comments and feedback to an intermediate draft, and Katherine Demacopoulos and Kathleen Tsaknakis provided critical project management and sound counsel throughout. My assistant Ginee Berardi made certain that book-related topics were a priority and were carefully orchestrated with my complex schedule.

Alexandra Jacobs was a tireless writing partner on the journey, and she always found a way to keep making progress despite other pressing obligations. The team from Wiley — Paul Levesque and Rev Mengle — honed and refined the final draft in preparation for publication. Finally, I am indebted to my wife Maggie Russell Berkes for her patience, support, and understanding for the many times I would drift off without warning into the realm of words.

The focus of this book is on achieving better business outcomes using technology, but businesses and their customers are only part of today's digital revolution. Governments, too, are working at every level to better serve their customers — citizens — through digital channels. To help further those goals, all proceeds from this book are being donated to Code for America, a non-partisan, non-profit organization whose mission is the use of technology to create a more open and collaborative relationship between citizens and government, and closing the digital gap between the private and public sectors.

Introduction

A new world is taking shape. It's a world dominated by digital engagement, where competitive differentiation is increasingly determined by a business's ability to embrace software-driven business models. It's a world where disruption is the norm and where long-established brands are renewing themselves by putting digital first. From taxis and travel to banking and retail, the foundation of almost every industry is undergoing profound change and revitalization. In response, businesses are remastering technology to thrive in a digital world by building software into their very DNA.

The rise of the application economy — like every other technological revolution — is also driving profound social, institutional, and economic change. There's no question, however, that as you look out at this shifting landscape, this new world is about one thing: pressure. Internal pressure from your employees; pressure from your board and your shareholders; pressure from your customers, partners, suppliers, and competitors; pressure to develop the capability to quickly deliver high-quality, consumer-like experiences on every device; pressure to fend off disruptors and to use technology to drive sustainable growth for your business. In other words, pressure to digitally transform.

While all the changes you face may seem daunting, they represent a unique opportunity to re-think, reshape, and reinvigorate your business. Throughout this book, we will offer simple, straightforward, and easy to understand starting points to remake your organization and give it the capability and flexibility to shape your future by enabling your customers to shape it with you. Think of these as guideposts for your journey rather than as a fixed recipe. Your success will depend on recognizing the behaviors and approaches that your organization can adopt to prosper in an environment of fast-paced, software-driven change.

Topics to be covered in greater detail along with some straightforward first steps include:

Building a Modern Software Factory and why older “waterfall” development methods don't work in a world where continuous connectivity and feedback shape the brand experience you give your customers.

Developing an Agile Mindset not only to reshape your software development process, but also to realign the flow of work across the business to increase both the velocity and efficiency of outcomes.

Building a Connected Business by using software techniques to enable rapid development, deliver new value to customers and partners, and capture changing market opportunities.

Securing Your Software to be able to meet your customers' expectations for safeguarding their data and ensuring the security of their digital engagement without compromising convenience and ease of use.

Enabling a Data-Driven Operating Model using two-way communication with customers to gain insights into their experience with your software and your brand.

Innovating for Business Success in a world where aligning a dynamic software investment portfolio with your strategic objectives determines your ability to deliver experiences at the speed your customers demand.

Depending on the current state of your digital strategy and investments, all of the preceding areas will be central to your digital transformation efforts at some point or another. This book is intended to provide a strategic view of technology and to enable an understanding around why these areas are more important than others as you embark on your digital journey.

If you are in the C-suite or a line of business leader, then this book will help you cut through the hype to drive your company forward. The chapters describe in non-technical terms how to build a software development capability so you can deliver the compelling digital experiences your customers are looking for. If you are an IT leader, you already know that becoming a digital business is an urgent priority in today's world, but where to start and how to engage others around your aspiration can be a challenge. This is a book you can share across and up the chain of command.

This book offers everyone a starting point for mapping out a digital journey and some helpful tips and advice for navigating the challenges along the way. If there's one thing that is certain in this increasingly fast-paced and changing world — you will need to master digital to succeed.

1

Remastering Your Business for a Digital World

Technology has an uncanny ability to generate hype and excitement as well as a degree of uncertainty. These days, there is an endless supply of headlines about digital disruption and software's ability to reshape business, industries, and even life as we know it. Startups appear out of nowhere and use technology to rapidly build share in markets that have been relatively stable for many years. Internet billionaires become celebrities seemingly overnight. Suddenly, companies are advised to transform themselves digitally or suffer dire consequences. As these dramas play out, consultants, advisors, and pundits rush in to join the fray. CIOs — and, increasingly, other C-level executives — are given charters to 'go mobile,' 'build apps,' 'get in the cloud,' and 'use big data.' It's easy to see why common sense thinking about what to do can get lost in all the noise.

The importance of software and digital technology in business is certainly real, but why is the situation today any different than in the past? After all, businesses have always been incorporating new technologies as they emerge in order to lower costs and increase efficiency. In fact, the use of computer hardware and software technology has been integral to the functioning of large enterprises since the mainframe over 50 years ago, and has spread to every business as costs declined and as technology became more accessible.

What's Different Now?

Until recently, the primary application of digital technology has been to *support* business rather than to actively *drive* business. Throughout this process, the technological footprint of companies has expanded exponentially, but most companies' business model, processes, and organizational framework have remained fundamentally unchanged. IT departments were built to increase the effectiveness of the existing pillars of the business — sales, marketing, product development, HR, and finance. The core mission of the classic IT department was deploying and operating technology to ensure the smoothest, most efficient execution of the various business functions. The business viewed software primarily as a productivity tool rather than as a *core component* of the business's DNA — an essential element, in other words, for creating value and driving sustained competitive differentiation.

The convergence of the Internet, mobile devices, and pervasive connectivity has fundamentally changed the business relationship with digital technology. Any business can now engage directly with their customers anytime and anywhere through Internet-connected applications and social media — and vice versa. The engagement is real-time, it is one-to-one, and there are no intermediaries between the company and the customer. Most importantly, it is this two-way engagement that is turning the traditional business world on its head. Today, digital engagement through software has become more than a basic expectation: software actually enables your customers to define your business with you through this new interaction model.

Redefining Technology's Role in Business

Using technology as the primary interface with your customers has profound implications for how your business is organized and operates. As the most direct conduit between a company and its customers, technology must now bear the weight of responsibility for the brand on its shoulders. A company's product may not be technology, but its brand will be represented, communicated, and judged *through* technology. Businesses must now be able to connect with and satisfy customers through software-based experiences. To successfully transform your business to enable this new channel to create value, you need to integrate software into the very core of how your business operates.

In order to build software into your business DNA, it's important to understand the drivers in a world where the digital experience you deliver *is* your brand. This new kind of experience is dynamic and continuous, and technology, instead of supporting relatively static line-of-business applications, must now deliver live, always-on software services. Where completeness of application *functionality* was once the prime technology metric, a continuously evolving user-centric *experience* is now paramount.

The traditional IT function that has served business effectively for decades is not equipped for these new responsibilities. Despite its limitations, many organizations today insist that IT operate in a dramatically different way and for a different purpose than the one for

which it was originally constructed. It's no wonder IT falls short, seems less than strategic, and struggles to deliver — it's been given a completely new job description, yet few new capabilities to fulfill it. But the answer isn't to simply enhance IT with new capabilities; it's about how to build a new operating model for business that puts technology — specifically software — at the very heart of the business itself.



The changes needed to embrace software's new role in business are transformational, not incremental. The methods, skills, and tools needed to build externally-facing, Internet-connected software experiences are radically different from those needed to incorporate software to optimize internal business processes and efficiency. Delivering a continuously evolving user experience with speed and scale has little in common with managing business application upgrades once or twice a year. Software must now be designed, built, and operated with a primary and critical focus on the business's customers and their experiences. Regardless of the market that a business happens to serve, this new use of software for direct customer engagement means that the business will also be providing software as a product that is consumed by its customers. In other words, every business must also become a software business.

Every Business Is in the Software and Service Business

This new software business is also a service business. The software experiences delivered to customers are live — they are constantly evolving and require continuous operation. Launching a website or an app is just the beginning of an ongoing relationship. Customers expect you to keep your sites and apps operating smoothly and securely, 24x7. User interfaces are just the visible portion — the final steps of a much larger journey in which design is a crucial differentiator in every aspect of the larger endeavor. Invisible but essential, a fabric of servers, databases, networks, and software must constantly be running in order to achieve experience delivery. For example, when users log into an app on their device, the heavy lifting is done by remote authentication services running as part of the operational environment. If those services aren't working properly, the user can't log in. Any such operational problems must be detected and fixed as quickly as possible. And customers expect new value to be delivered as their needs change. Delivering externally-facing software experiences is an ongoing, constant operation.

The experience that customers have with a business's apps will have a profound impact on the company's brand. Customers will judge companies through their experiences with those apps. Can they get to the information or functionality they need quickly and easily? Is the user interface simple and intuitive? Is the software responsive and free of defects?

Is it available on all of the necessary devices and web browsers? Does it integrate seamlessly with social media channels? How these questions get answered will determine the success or failure of your brand.



In this new era, the ease of access to brands through software has a flattening effect on competition. It's no longer enough to compete within narrow product categories. Competition now requires delivering the best possible software experience regardless of product category or even industry. Software has leveled the playing field by allowing old, new, large, and small companies to innovate rapidly and to continuously redefine and reinvent themselves as their markets and the needs of their customers change.

Customer Engagement through Software

Digital engagement changes the relationship with customers and is a powerful mechanism for sensing and responding to unmet needs. Digitally interacting with a customer through a live, Internet-connected application allows a deeper understanding of that customer as an individual because the flow of data is ongoing and bidirectional. Any aspect of the user journey through the software experience can be recorded and analyzed to gain insights into user preferences, behavior, and needs. The interaction is one-to-one, and each customer is understood as a “market of one,” thus enabling experiences to be individually tailored. Critically, data gained from such interactions can be the source of deep insights to inform the business across every dimension. The feedback loop between the data gathered through digital engagement and the insights delivered back into the business is one of the most important aspects of this new way of operating. In fact, this loop is the core mechanism for a completely new kind of customer intimacy — one that gives you the ability to shape your business in close partnership with your customers.

The software-driven changes we are witnessing are just the beginning of a large-scale restructuring of how businesses operate. Software-based engagement removes the barriers between businesses and their customers and allows intimate, personalized connections; no one is just a number. Because software is fluid, it allows digital engagement to be dynamic and responsive. And the live nature of today's software experiences makes it possible to understand — and respond to — customer needs, desires, and behavior in real time. But these new capabilities can't be cleanly partitioned along traditional business functions. Who is responsible for which features to prioritize? Sales? Marketing? Technology? Product? Who determines what customer data should be collected, and which insights would be the most valuable? And who decides what actions those insights should drive? The division of labor and organizational structures honed in the last century were never designed for real-time input or decision-making. Not surprisingly, they fall short.



Software can express a limitless number of ideas and can help answer an equally limitless number of questions, but software's potential can only be realized by working beyond traditional business functions and activities. In a real sense, software must be the hub around which all aspects of modern business must come together to realize the business's full potential. The technology organization must be able to create, build, and operate software, but it must have full support and involvement across the entire spectrum of the business to do its job effectively. Any gaps in collaboration around the feedback loop with customers will mean lost engagement opportunities, and may damage the software experience and even the brand itself.

PUTTING TECHNOLOGY – NOT IT – AT THE CENTER OF YOUR BUSINESS

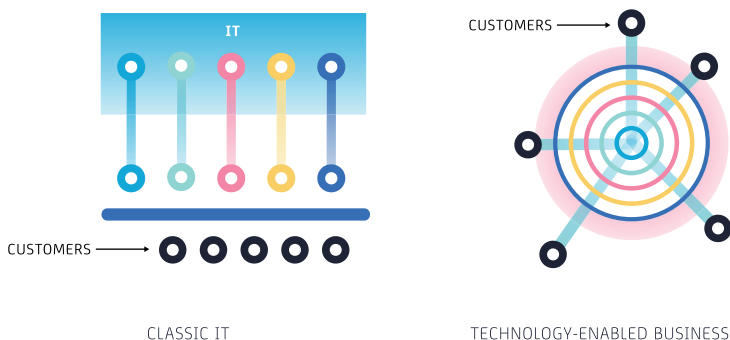


Figure 1-1: In the past, it was IT's job to support slow-moving line-of-business applications. But in a world where satisfying customers through software-based experiences is paramount, it's critical to integrate software into the core of how your business operates — in other words, your business DNA. Putting technology — not the traditional IT department — at the center of your business will enable you to continuously deliver software experiences to your customers via live, always-on services.

Building a Modern Business with Software at the Center

Incorporating software development as an integral part of business may seem like a daunting task. The first important step is to recognize the fundamentally different nature of software's role in business today. This is not simply a matter of traditional IT with some added responsibility. The ability to deliver customer-facing, software-based experiences is

an entirely new capability that you may need to build from the ground up. Your focus needs to shift outside the enterprise to the customer. You will need to create software execution competency — a kind of modern software factory — that will enable you to continuously deliver software experiences at scale to your customers. You will need to revisit your priorities and shift your investments. You will need to enable a more efficient and collaborative operating model optimized for both speed and customer value. And using digital engagement, you will need to create feedback loops so you can provide your business with a steady flow of actionable insights.

The good news is you don't have to change all at once. Your goal should not be perfection; transformation is a messy process. Your goal is to create the right environment for continuous improvement, discovery, and experimentation. Your mission is to create a framework to enable the possible and to embrace open-ended change. The days of discrete projects with neat, linear timelines from start to finish are gone. So, too, are projects that deliver features and functionality — at great expense and strain on precious resources — that nobody really wanted in the first place. Your goal is not transformation; that will happen naturally along the way. Your real goal is to uncover the clarity that comes from knowing your customers through the software you build. If you adopt a simple strategic and technological framework and identify some obvious first steps, you can begin to adapt your business to an increasingly software-centric world and set a direction that will continue to guide you into the future.

Welcome to Your New Operating Environment

In the world that is to come, your operating environment is going to move away from a rigid and hierarchical model with limited customer engagement to one that is more fluid and open, constantly seeking both customer and employee interaction. Your old system will be pushed to its limits as linear business practices intersect with nonlinear technological change, and as this happens, you must be developing the muscle and flexibility that will help you transition into the future. To prepare for this journey, it's important to grasp some key emerging trends that will help you adapt and shape your business with software at the center.

Shifting boundaries

First, the boundaries between the enterprise and the outside world are changing. The old operating models that are now crumbling were based on a separation of activities between the inside and outside of the enterprise, managed by different sets of rules. Control in these hierarchical models was derived from the notion of a distinct wall between the enterprise and its external environment. Technology has changed all that. We see the signs of this shift in things like rogue IT — where employees go to the market for the tools they need rather than relying on traditional IT departments. It is also the catalyst for the social enterprise — where social media and customer intimacy have become a

new direct channel for business. As a result of this process, successful organizations in the future will likely look very different from the ones of the past. And the longer any organization holds on to old operating models, the harder it will be to transition going forward.

Redefined identity

Second, technology-enabled experiences will continue to become richer while challenging our conventional notions of personal identity. Ubiquitous device connectivity, artificial intelligence, augmented and virtual reality, and voice and sensor-based interaction combined with user-centric design will create experiences that capture highly granular and diverse sources of data from individuals revealing both *what* they are experiencing and *who* they are as individuals. As more and more data is collected, human beings will increasingly be defined by their individual and collective digital experiences. Personal identity itself could consist of the technological net sum of individuals' interactions which may be externalized, shared, or made to serve as digital proxies for our real selves.

Exponential innovation

Third, our approach to decision-making will change. The human brain — hard-wired to detect patterns and to learn based on them — will be augmented by exponential technology innovation. As adept as we may be at learning, we are ill equipped to find hidden patterns in billions of pieces of data, and there are limits to the usefulness of the experiences we build without the ability to learn and adapt. Although artificial intelligence is sometimes cast as a sinister force that replaces our own intelligence, it can certainly be used to augment and amplify our own. At a minimum, artificial intelligence techniques such as cognitive computing and deep learning can take some of the guesswork out of important decisions by extracting insights from an almost limitless supply of data. In the new world of the customer-defined business, a wide array of tools and techniques will not only reveal unmet needs but will also light the way to future opportunities.

What's Next

As we've outlined in this chapter, there is sweeping technological change underway in the world that will challenge the notion of the traditional enterprise and one of its oldest functions: the IT department. The ascendancy of customer experience in a user-driven world is driving technology into the very heart of every company's business model and becoming one of the primary determinants of the success of its brand. The road to the future involves the creation a modern software factory that can both design and continuously deliver experiences in an always-on world. Modern software development competency must be deeply rooted in customer intimacy. That intimacy is gained through digital feedback loops, applied analytics, and an agile and efficient software development process that is constantly translating customer need into delivered experience.

As you read the following chapters, keep in mind that there is no convenient shortcut or precise, prescriptive plan to achieve the type of transformation that is the topic of this book. The journey ahead of you is multi-faceted — there isn't a single element that by itself will achieve the outcomes you seek. Although tools, techniques, and technology play vital roles, culture, people, and mindsets are just as important, if not more so. Don't get lost in technical details or fine points, and focus instead on key themes that in combination create a new system for engaging with your customers and building the future of your business.

2

A Blueprint for Your Modern Software Factory

If the traditional IT model is not equipped to drive a digital business, then what model is? That is the question at the very center of all digital transformation initiatives. Every company faces intense pressure to come up with the right answer for ‘going digital’ to inoculate against disruption. But arriving at a model capable of serving the needs of your business in the digital era requires a fundamental reassessment of technology’s role.

The truth is that digital technology has been a focal point for business for over 50 years. It’s grown horizontally, traveling out along the pre-formed paths of traditional organizational structures, and has grown vertically within the functional silos of HR, finance, product development, manufacturing, and marketing — yet it has rarely managed to make the leap between those silos. In fact, much of the digital technology you use now was actually born in the 60’s and 70’s. It became the tool of choice for accelerating the manufacturing and conglomerate boom of those years, and, after that, it was used as a lever to drive the outsourcing and cost-cutting waves in the 80’s and 90’s. It has grown up with your organization and, today, it’s wrapped around most of your business processes, is expressed in your culture, and is a deep contributor to the rigidity and stasis that makes it difficult to drive change — or even know where to begin to make that change.

The way your technology evolved to its current state is precisely why it is impossible to simply layer on new approaches — agile or DevOps (Development Operations) methodologies, for example — to achieve real change. It’s like trying to plant a new tree right next to an old one — the deep roots and shade of the old tree will starve the new one of nourishment and light every time. Two of the hardest things to change in any organization are

technology and culture, and together they can work against meaningful transformation. Yet to be able to put software at the center of your business in earnest, you will need to change both. You need more than incremental modifications to what you have today; what you really need is a different starting point.



Today, your customers may be your biggest disruptors and are increasingly your most important partners in the redesign of your business through software. If delivering a high quality, seamless digital experience to your customers is to be the foundation of your brand, then reengineering everything with that goal in mind is your best chance for transformation success. Your customers' demands place new challenges on traditional IT. With its emphasis on stability rather than rapid evolution, a small-scale, old-style "custom-shop" approach simply can't deliver the innovation that customers demand and it can't create the scale of software output required to satisfy the needs of a digital business. The process of software development and delivery must be transformed to achieve the key attributes of any modern production operation: high throughput, consistent quality, extensive automation, and resource efficiency. In other words, what you need is a modern software factory.

Optimizing for Continuous Development and Delivery

How well your software factory works will determine your success in the market. New approaches and methodologies such as agile and DevOps are essential to driving speed of innovation and responsive, continuous engagement with your customers. A winning software factory must be completely immersed in the context of the customer journey — transforming their needs into proven delivered value. That is one of the core promises of agile. Ultimately, the factory will enable you to deliver four things to your customers:

- **Trust.** Your customers will feel secure enough to share their sensitive information with you.
- **Value.** You will give customers the value and high quality user experience they expect.
- **Speed.** You will be nimble, responding to and delivering unmet customer needs quickly.
- **Reliability.** Your apps and services will work always, anytime, anywhere.

Core Operating Principles of Your Modern Software Factory

To achieve the goals outlined above, your operating model will need to adhere to three core principles. Your factory must be:

- **Automated.** You must use automation wherever possible to drive out errors, build in predictability, enable high throughput, and free up resources to focus on higher-value areas.
- **Agile.** You should use approaches such as team-based agile development and agile frameworks that scale to add ever-higher levels of flexibility and responsiveness around your business model and strategy.
- **Insight-driven.** Your software is designed to give you access to a diverse and constant stream of feedback that smart analytics can then turn into insights you can use to adjust and improve your business performance.

With a singular focus on the customer, these core principles will lead you to adopt new methodologies and approaches as you bring your software factory online. This will include things like human-centered design, behavior-driven development, multi-variable testing, and deep personalization to help create richer and more meaningful customer experiences across the entire digital journey. This new toolset will be driven by robust analytics and applied heuristics that are real-world (and sometimes real-time) reflections of your existing and potential consumers. These approaches adopt and expand the concepts of continuous, omni-channel delivery — ensuring that every customer experience is consistent, integrated, and connected across every digital and physical channel.

Two Software Cycles: Experience and Service

Your new operating model has two main “cycles” — each with its own unique challenges and opportunities. Because your end users are consuming live experiences, your technology ecosystem has to perform two critical functions. First, it must give you a way to ideate and build software as an *experience* that represents and reinforces your brand. This is not about digital commerce; if it were, you could just build a simple e-commerce portal and be done. This is about digital engagement, and everything you do to foster that engagement is integral to the experience creation part of the software cycles within your organization.

Second, you have to find a way to deliver your software experiences as a reliable *service* — that means 24/7 with utter reliability, high quality, and the capability to provide continuous updates along the way. This cycle is all about uptime, availability, responsiveness, and feedback on the performance of your experiences. The new three-second-rule — the fact that users will abandon your app in three seconds or less if the app fails to come to life in that time — puts tremendous pressure on your organization's technology ecosystem to get this part of the software cycle right.

BUILDING A MODERN SOFTWARE FACTORY

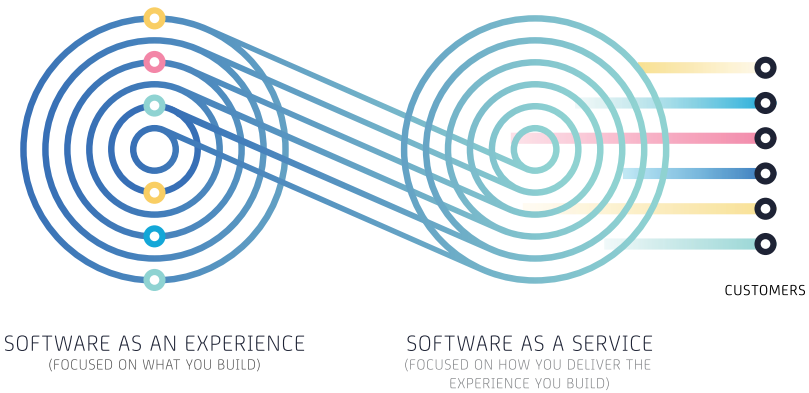


Figure 2-1: A winning software factory must be able deliver engaging software experiences *and* deliver them continually and at high levels of quality. The technology in your organization needs to be structured to enable both “Software-as-an-experience” and “Software-as-a-service” to drive brand value.

Software-as-an-experience

The technology cycle that creates the experiences you deliver is what most companies would think of as their “dev shop.” Although most people envision aging infrastructure and older software when they think of legacy IT, the heritage of IT operations is reliability and predictability, qualities that should be preserved in the modern software factory. When the primary function of technology in business was to support core back office functions — classic IT, in other words — software applications and development were inwardly focused to support those business processes. With an emphasis on managing and limiting risk, there was little room or need for fast-moving software innovation. Apart from business-specific customizations and routine maintenance, software applications were infrequently updated or changed.

The business landscape has changed radically. Rather than leveraging software technology merely to do more of the same — only better, faster, and cheaper — businesses now depend on a constant flow of new software services to connect and engage with customers in fundamentally different ways. To that end, building and optimizing a streamlined software delivery factory is a serious undertaking. There are many manufacturers (teams) to keep productive, complex part-lists and components (features, software code, builds, and configurations) to coordinate, and suppliers (technology vendors) to manage. Compounding the challenge of building a new software factory is the fact that you’re not starting fresh. You have existing teams operating in traditional discrete functional silos, and processes and technology designed to deliver software on a yearly rather than weekly or daily cadence.

Both agile and DevOps methodologies can play key roles in building and optimizing your new software factory. Agile is an approach for organizing customer-focused development based on the efficiency of small, integrated teams working together. DevOps is about aligning people, process, and technology to enable a leaner, more integrated approach to operating your application development and delivery pipeline. While the terminology can be confusing, both agile and DevOps are ways to remove roadblocks to rapid and continuous delivery of high-quality software. They work together to combat the pitfalls of old-school software development, such as:

- Software defects discovered late in the lifecycle, or worse, after the software has been made available to customers
- Loss of any gains in speed of development due to long delays in the process of releasing the software to customers
- Impeded progress due to lack of sufficient access to development and test environments needed to develop and test code
- Lack of visibility into the root causes of problems resulting in inefficient use of resources and organizational confusion

The combination of agile and DevOps practices aims to address such issues head-on by stressing collaboration and communication across all the functional teams and disciplines involved in designing, building, operationalizing, and running software delivered to customers. The result is a software factory that streamlines the sequencing and orchestration of building, testing, and deploying software, and enables the systematic and repeatable delivery of ever-improving software experiences to customers.

As in any modern factory, automation is a key technology for creating efficiency. Although automation of your existing approach can drive incremental improvement, you will need to look more broadly and include transformational technology that can enable you to do some things in a completely different way to maximize automation's potential. Examples include:

- Tools to simulate real-world conditions so that you can automate testing and optimize code as early as possible in the development process
- Tools for defining and managing software requirements to ensure high-efficiency collaboration that delivers the desired features and functionality
- Tools for measuring code quality and security
- Tools to record and analyze data to gain insights into how your customers are experiencing the software you're delivering

DevOps and agile methodologies improve the process of software delivery to optimize the performance of your software factory. With them, organizations can reduce the time, risk, and expense of software delivery by frequently releasing small batches of new value. These smaller batches make the corresponding code changes easier to track through the process and improve the ability to accurately trace — and quickly address — defects along the way. Faster, more frequent software releases with higher quality will mean happier customers and a better business outcome.

Software-as-a-service

Digital consumers have come to expect the same kind of reliability from Internet-connected applications one used to get from the phone company — where everyone knew that picking up the phone meant hearing a dial tone 99.999% of the time. Such high expectations have now been transferred to web-based apps, which are expected to operate reliably, all day and all night, every day of the year. You must think about building and operating your applications as if you were a utility. When a customer clicks on a link to your web page, the software behind that page needs to be up and running. When customers try to log in to your app to check on their account balance, all of the software that must run to authenticate the customer needs to be fully operational.



What most people think of as the app — the user interface and the software that gets downloaded and installed on phones and other devices — is only one small part of the application. Much of the work to bring applications to life happens far away from the device, in data centers connected to the Internet running software that — although remote — is an intrinsic and essential part of the experience.

The consequences of failing to provide the essential services that power your apps are far higher than for any internally-facing product or service. While it may be embarrassing and inconvenient to have a temporary outage for your corporate email systems or even financial systems, the stakes are much higher when external customers are relying on your software services. How useful is the app if the user can't log in? How useful is the app if users can't access their bank balance, or music, or photos? While Twitter's "Fail Whale" was a whimsical way to apologize for service outages, it also became a symbol for users' frustrations when they couldn't access the service.

The day you launch your app is the first day of a de-facto, ongoing relationship that you establish with your customers. Just like electricity or Internet connectivity, we take the software services that power applications for granted, yet these services are the foundation of the application economy.

What Should Your Software Factory Look Like?

Your software factory may be busy, but it should never be a black box or a tangled mess — it should always be well defined and well understood. It should include everything that it takes to get from ideation to product or service, and ultimately, brand experience. Taking a page from quality-assurance philosophies, it means ensuring anything that is implicit is made explicit and measurable.

There are a few common tenets of a well-built software factory:

- **Clarity from design all the way through to manufacturing.** What's developed and tested always reflects back to customer value and need
- **High automation.** Your factory needs to be repeatable, measurable, and predictable
- **Constantly running.** Continuous everything — integration, building, testing, and deployment
- **Always available.** Every part of the factory is available 24/7, including upstream- or downstream-dependent services or APIs
- **Data gathering and analysis everywhere.** Telemetry is built in from the beginning
- **Resiliency.** Single points of failure are mitigated, and software production can quickly be adjusted to accommodate new scenarios

These tenets are aimed at keeping your software factory operating efficiently at full speed far into the future.

What Will Your Software Factory Really Look Like?

Most companies haven't had the luxury of starting fresh like start-ups can. Young, "cloud native" organizations started with a clean slate, with no legacy backend systems, no millions of lines of existing code written over many years, and no complex dependencies on third-party systems or complex infrastructure. That said, some of the best software factories currently churning out truly compelling disruptive innovations are traditional enterprises — those that have made the changes needed to reshape their digital presence.

Neither agile or DevOps, or any other methodology will ever be a magic bullet. It's important for you to understand your operating context to inform the choices you make in building the software factory best suited for your situation. If your business is new to customer-facing software development, understanding some key points will help allay fears about not being perfect and enable you to aspire to ever-increasing proficiency:

- Every software factory will be different in composition.
- Factories will be heterogeneous and may include many technologies and platform components; standardization is preferable but not always necessary (optimize pragmatically).
- Focus on measuring both velocity (speed and throughput) and control (consistency and predictability).



No software factory is perfect. Forget perfection; what is truly important is your ability to prioritize, recognize areas for improvement, and respond accordingly on a continuous, ongoing basis.



Getting Started

Key to your transformation is understanding that your new goals are fundamentally different from those that may have guided your software development practices before. Your primary focus is now outside the company and on your customers. What you build and deliver is not merely an adjustment to your business but an integral pillar of your business. You will need to rethink and redesign your technology function to be able to design, build, and deliver experiences rather than mere utility. The first steps in this process are essential ones:

1. Document what you have today.

Before you embark on changes, you must first analyze the software capabilities you have. Is your current development mechanism well understood? What is the process and flow that an idea takes on its way to being a product or service? What are the interfaces between the different stages of development? It is likely that what you have in place evolved organically, responding to ad-hoc needs rather than through intentional design. In order to identify weak spots and friction points, your current processes must be documented and articulated. And while it's important for you to understand your current state, don't fall into the trap of merely optimizing your current development operation. Such local optimization will certainly result in some improvement, but you need to be thinking about optimizing globally to ensure that the constraints across the entire process are addressed.

2. Design your future factory.

Just as it is important to understand where you are now, it is even more critical to know what you want to build in the future. What is your ideal state? If you had a blank sheet of paper, what software factory would you design? What is the fastest path between ideas and tangible experiences? Where are the necessary connection points? Optimize for throughput, keeping in mind that you need to balance both software-as-an-experience and software-as-a-service cycles to achieve maximum customer impact. Don't attempt to do this in a vacuum — you need representation from all disciplines to collaborate on the architecture. And throughout the design process, keep your focus on outcomes — the things that matter to your customers — rather than internally focused factors.

3. Sequence and integrate your needed changes.

After you've captured your current state and architected your ideal future state, you can start mapping between them. Where are the biggest gaps between how things are today and how they should be? Which changes would not only be relatively

easy to make but would also yield significant results? Which ones will require more time, planning, and orchestration? Some changes will require one-time, large-scale disruption (for example, moving an entire organization to agile), and others can be more gradual or compartmentalized (such as shifting to DevOps). Keep in mind that the goal of your changes is to drive continuous business improvement requiring automated information flow within and across processes and teams as you move forward. For example, information on application usage and performance should be captured and forwarded to product managers and developers, helping them prioritize functional enhancements based on customer preferences.



Building a modern software factory will require time and dedication but will give you the high throughput, consistent quality, extensive automation, and resource efficiency that your business demands.



Partner Perspective

Williams-Sonoma rethinks “IT”¹

As more companies and organizations undergo digital transformation, there's an increasing debate about who owns the technology stack. Not that many years ago, the answer would have been “IT,” hands down. Today, that answer can be a bit murkier as IT, marketing, and line-of-business owners can all stake a claim.

Iconic lifestyle brand Williams-Sonoma has nipped this argument in the bud by combining the top technology (CIO) and marketing (CMO) roles under one executive: John Strain, executive vice president, chief digital and technology officer. “Some people think it’s an interesting mix to bring technology and marketing together. We see it as a true competitive advantage,” Strain explains.

Strain and his team oversee the technology and marketing for Williams-Sonoma’s namesake brand as well as a number of other high-end retail properties, such as Pottery Barn and West Elm. The team’s goal is to create a customer experience that’s consistent across store, website, and mobile. Says Strain: “It’s all about being customer-centric and letting them shop through whatever channel makes the most sense for them.”

There is no such thing as a “technology project” at Williams-Sonoma. Everything they do is a business project, as every project has a business value. Strain and the organization drive these projects based on the merits of the business results they’re going to be able to provide. He explains that the company looks at projects with both

(continued)

(continued)

a business and technology perspective to help prioritize projects. They look at time to value, payback, and ROI along with normal IT metrics. The ROI push and the business story help land projects on the priority list.

For instance, personalization is huge for the company. Williams-Sonoma uses the vast amount of information it knows about its customer buying and browsing habits to tailor what they'll see on the website. It uses trigger-based emails to deliver personalized offers based on a user's history or interests. The trigger emails represent only 6 percent of the company's outbound emails, but they drive 40 percent of email revenue. That number continues to grow by double digits each year.

The IT staff uses agile methodologies to speed its development efforts and bring greater value to the company. Prior to adopting agile, the company had one major release a year with a few minor quarterly updates. Now, the company has reduced its release time to the cadence appropriate for its business — three weeks — with feature-rich updates rolling out to all seven of its major brands far more consistently.

Williams-Sonoma's digital transformation efforts don't end at commerce — they impact every aspect of the business. One area the company is trying to improve revolves around delivery of large, custom items such as furniture, which is not as easy to move as smaller parcels. If a customer orders a custom couch, it's made at a factory in North Carolina and then needs to be transported to the customer's house and set up. Doing that in a seamless way is not easy and requires a lot of coordination.

"We see tremendous opportunity for us to be able to take out costs, improve efficiencies, and, at the same time, improve customer service levels. Getting the supply chain right will be a competitive differentiator," Strain says.

Williams-Sonoma may be known for its beautiful print catalogs, but digital transformation enabled it to build on its foundation to personalize the experience customers have based on new insights. Now, creative decisions come with more analytics, testing, and vetting before being rolled out. Says Strain: "We're right at that intersection between lifestyle merchandising and analytics, and that blend and that balance between the two is really critical for us."

3

Building Your Software Factory

In building your software factory, your primary goal is to create the capability necessary for translating ideas into live experiences. The question then is how to translate those ideas as efficiently as possible into experiences that customers can see, hear, and touch. As companies become more proficient with highly iterative software development and collaborative approaches like agile, they begin to close the distance between idea and experience. They learn to quickly turn an idea into a theme, a story, a design or wireframe, and then a set of requirements. Within weeks, development teams can work through those requirements to create a working prototype or even the initial version of a market-ready product. This acceleration is palpable across development teams and is really what agile is all about.

Delivering Software Continuously by Optimizing ‘Below the Line’ Activities

While the pace of design and development typically accelerates rapidly in the early stages of building your software factory, the cycle time for releasing the software to customers often fails to keep pace. There are many reasons for this disconnect, including:

- Infrastructure delays and the high level of effort required to set up test and development environments
- Complexity of aligning software testing with changing requirements and partially complete functionality

- Integration dependencies between different software components required for implementation
- Overhead of managing “stacks” of infrastructure, middleware, and software platforms
- Orchestration across an alphabet soup of software development lifecycle tools in the chain

The “secret sauce” of your software factory will involve building a well-defined, highly automated process that allows for a constant flow of software releases designed to keep up with your pace of innovation. You can think of this as “below the line” technology, skills and processes that give you more time to focus on “above the line” software — the experiences that you build for your customers to keep you ahead of your competition.

Increasing the pace of delivery requires rethinking the sequencing and the serialization of work. In a traditional “waterfall”-style approach to delivering an IT product, the progression from one functional team to another is clearly articulated with each team — a specific set of self-contained responsibilities, expertise, and deliverables is associated with each step along the way. With this model, deficiencies that are found towards the end of the chain are costly and time-consuming to fix since doing so requires restarting the entire sequence. Furthermore, any problems encountered along the way to delivery can lead to animosity and finger-pointing between functional areas, which takes time and energy away from solving the technical challenges that inevitably arise with any complex software endeavor.

A potentially messier but faster and more efficient approach to software development is to integrate all aspects of software delivery as early into the development process as possible. In the DevOps model, everyone involved with developing, deploying, and operating a set of features is on the same team. Although there is a very well-defined process for delivering software, everyone on the team participates, and when something doesn't go as planned, it's the team's responsibility to address the problem; they're in it together.



Writing software is challenging, and it's impossible to deliver perfect code 100% of the time. Missing or incorrect syntax in the code, faulty range-checking of data, failure to handle unexpected input gracefully, incorrect interpretation of the desired outcome: the list of the many things that can go wrong with software is virtually endless. Consider the fact that, with his Halting Problem proof in 1936, Alan Turing demonstrated the impossibility of determining the correctness of software. Delivering great code is part art and part science, and a DevOps approach grounded in collaborative teamwork and continuous, rapid integration of new code into the production cycle can significantly improve both speed and quality. The earlier that code can be run and tested using real-world scenarios, the earlier any errors can be caught and fixed.

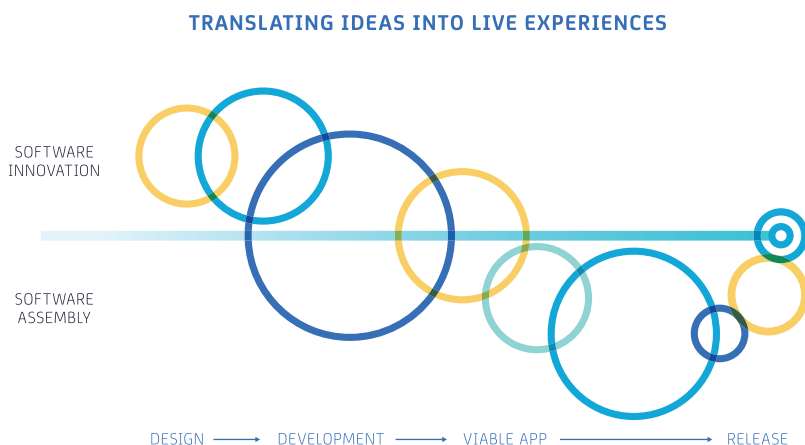


Figure 3-1: The success of your software factory will depend on your ability to build a well-defined, highly automated process to deliver a constant flow of software releases. This “below the line” set of technologies, skills, and processes drive efficiencies that free up time and resources for work “above the line” — the ideation and innovation that create the experiences themselves.

Building Your Factory Team

The essence of your software development capability comes from the collective power of the individuals you bring together to form your software factory team. Addressing the “need for speed” that is part of all software development cycles will require a combination of the right tools and technologies as well as people with the skills to use them effectively to architect and operate a friction-free software production line. You will need to create a highly automated system with the right people and processes to integrate, test, deploy, and operate the software you develop. In parallel, you must also acquire the capability and skills you need for “above the line” software innovation — all the pieces that make up the experience delivered to customers. This includes everything from user interface and interaction design to software architecture and development.

The following list summarizes the key considerations you need to keep in mind when building the team you need:

- **Some of the roles and titles you currently have in place may give you a false sense of security.** You must reevaluate and refresh your roles and job requirements carefully to ensure that you can retain and attract the skills needed to design, build, and operate customer-facing software experiences.
- **You may need to bring in expertise that you didn’t previously have because there was no need.** Product designers, software architects, and app developers are

common skills gaps in enterprises. The business of building software experiences is a radically different endeavor than building line-of-business tools, and you need to look beyond filling jobs that have been defined by the needs of classic, internally-focused IT.

- **When hiring, go for depth *and* breadth.** There's a move today towards acquiring higher-order problem-solving acumen in combination with core math, science, or coding expertise — the “T”-shaped employee with deep domain-specific expertise who is also able to think broadly across multiple problem domains. The type of innovation you need requires creativity, lateral thinking skills, and the ability to operate with high levels of ambiguity.
- **Your factory will constantly be changing and evolving — sometimes rapidly — in response to customer need.** Identifying candidates who can embrace change and who can thrive under uncertainty is of equal importance to obtaining candidates with specific technical skills.
- **In the fast-paced software world, existing skills become outdated quickly, and the ability to change and learn is essential.** Hiring against a list of technical specs may be useful to fill a specific need right now, but your list will change rapidly over time. You must also hire for the ability to change, adapt, and most importantly, continuously learn.

As you build your software factory team, you will need to find the appropriate organizational framework to support it. You likely already have an existing technology organization with responsibilities that must continue to be addressed. Do you start a completely new group? Do you refactor or repurpose existing groups to align to a new, broader mission? Will your new software factory operate in parallel with your existing technology efforts?

How you answer these questions will depend on many factors, including business need, existing capabilities, and company support. You may be best served by starting with a smaller, more focused effort working in parallel to existing groups, or by driving sweeping changes across the entire organization, or with some hybrid approach. Regardless of what you settle on, you must have a clear picture of your ideal end state and the steps you will take to achieve it. Keep in mind that the full power of methodologies such as agile and DevOps can only be fully realized at scale when applied across the entire organization. Don't stop until your transformation to a new way of operating is fully complete.

To attract and retain the talent you need to achieve your goal, you will need to review your compensation and ensure the right roles are defined and enable collaboration in the right ways to drive value. You won't just be competing with other enterprises for your talent. You will also be competing with start-ups and software-native companies. Do you have the right environment to attract designers, architects, and developers? Do you have the appropriate physical space for teamwork and collaboration? Have you streamlined internal processes, provided the right tools, and removed friction points and roadblocks to enable your team to focus on this mission?



Traditional approaches for building enterprise technology teams can't fulfill the new capabilities needed to build rapidly changing customer-focused software. Although your business may not be software, you will need to incorporate best practices and methodologies used in the software industry to essentially build a software company within your business.

Common Obstacles to a Smoothly Operating Factory

In computer applications, parallel processing — working simultaneously on different aspects of a task — can definitely accelerate throughput of work, but it requires careful communication and coordination to achieve its potential benefits. This is also true when the concept is applied to your software factory. It is possible for teams to work simultaneously on different aspects of the software product to quickly build new functionality and to identify and address design flaws, security gaps, performance problems, and a host of other issues well in advance of going live. For such cooperative work to succeed, however, careful orchestration between the elements of work and the sometimes-fuzzy line demarcating development and delivery of the customer experience is required to ensure that no element of work is blocked and that everyone is moving forward in unison. This way, you can spot performance problems well before development of a feature is complete, quickly adjust the user interface during the initial stages of software design, or change the way a feature works based on early feedback.

Achieving the full benefits of parallel workflows moving together at full speed will require addressing a number of key challenges. The following list summarizes some of the obstacles you may need to overcome:

- **Lack of automation.** To achieve desired development velocity, teams must be able to deliver multiple software “builds” — new versions of a software product incorporating the most recent changes and additions — every day. Each of these builds needs to be tested and validated in order to quickly move them through the software delivery factory.

This goal can be compromised when teams have to manually create and configure the technical environments needed, perform significant amounts of hands-on testing, or deal with unanticipated differences in how the software works as it moves through different environments — the “it worked fine on my PC” problem. This can cause delays and pressure to implement short-term workarounds that introduce further fragility in the system that results in costly rework later.



Partner Perspective

Lockheed Martin uses DevOps to transform software development¹

Changing the software development process at a large company is a process in itself. For aerospace, defense, and technology giant Lockheed Martin, the adoption of next-generation methodologies like DevOps is as much about organizational and cultural transformation as it is technology. And it doesn't happen overnight.

"We've been on this evolutionary path to do more from an efficiency perspective," says Liz Michaud, Lockheed Martin's Director of Business Applications. Progress has come by focusing on the ways people work together and the skills they bring to the job, and this approach is starting to pay off. "We have had some thrilling success applying the DevOps model to some of our newer applications," she says.

When Ms. Michaud took her current job three years ago, Lockheed Martin had already begun the process of breaking down development and operational silos that in some cases went back decades. But there was more to be done to modernize the approach of her IT operations staff, which includes about 800 people who support some 1,600 internal applications across the sprawling \$45 billion company. The operations team has responsibility for everything from manufacturing and finance to supply chain and human resources.

One key organizational change was moving to a more centralized IT operations structure with a view of the entire enterprise, and then making sure the IT professionals were closely aligned with the business areas they supported. For example, a senior IT manager will now work directly with a program manager on the business side, with the business team defining what it needs from IT to fit its strategy.

Getting serious about DevOps means finding the right people to do the job. Ms. Michaud identified several "pockets of excellence" on her team and assembled a core group who do what she laughingly calls "DevOpsian" work. Currently, they have about 20 programs running across the company. Cloud-based applications and automated testing are the primary targets.

Success is measured in the quality of added functionality as well as the speed at which it is produced. New metrics are being introduced for comparing the speed of DevOps to traditional baselines — but also in cultural change. There is evidence that this cultural change is happening; it is anecdotal, but compelling. One application for generating proposals proved so popular that it began spreading beyond its original user base before senior management had formally blessed such an expansion. "People said, yeah, we're going forth with this — it is almost organically happening, and that's a telling sign about the ability to meet functional needs."

- **Poor orchestration.** In the modern software factory, autonomous teams are empowered to use their own sets of tools for areas like software change management, version control, provisioning, configuration control, and release management. Some combination of these tools often don't work together, and orchestrating all the interdependent functions in order to coordinate releases requires manual processes or scripting which can take more time than the tools are meant to save.
- **Bottlenecks and delays.** Many organizations are plagued by delays due to lengthy handoff protocols and a lack of task and activity coordination. Delays are harmful by themselves, but can also encourage other poor practices impacting quality. For example, developers must “batch up” their code changes to work around long release delays, making it much more difficult to coordinate those changes with others in the system and leading to higher likelihood of problems when it's time to integrate all of the queued-up changes.
- **Overreliance on operations.** Many organizations have placed a lion's share of responsibility on the operations team who are often called in at the end of the development cycle to ensure that applications are production-ready. Because this happens late in the release process, key insights that could have influenced major architectural, development, and system design decisions are missed. The operations and development teams must collaborate throughout the entire development lifecycle to ensure that development informs operations and vice versa. By the time low-quality code hits production, it's too late.
- **Testing silos.** Testing software requires knowing exactly what it's supposed to do. Problems can arise when separate groups within the organization create differing descriptions for how software functions should work, leading to conflicting and inconsistent interpretations of correctness. Those differences are then multiplied across different tools and tests. A core part of the problem is the very notion that testing — assuring the quality of software produced — can be the responsibility of dedicated testing teams. Throwing code over the wall to a testing team to ensure that the code is of high quality and performs as needed is an impossible task, yet this is how many development organizations operate.



Getting Started

Accelerating the pace of your software delivery is vital for achieving a rapid flow of new value to your customers. There is no silver bullet — no one-time program that you can administer and then be done. You have to be ready to approach creating this capability just as you must approach your software experience delivery — as a process of constant iteration and improvement. These are some guidelines to apply and steps to take in building and optimizing your software factory:

- **Integrate teams.** Modern software development is a team endeavor and requires a high degree of integration between people and processes. Instead of having sharp divisions between design, development, testing, security, operations, architecture,

and other disciplines, you must work to make those boundaries as permeable as possible. Everyone has a role to play and brings specific skills, but the focus must be on outcomes and adaptive problem solving. Adopting agile practices will help you ground your organization around small, cross-functional teams working together to deliver value to your customers.

- **Integrate quality.** Ensuring software quality must be the responsibility of the entire software factory from design all the way through operations. While software testing is a specific and necessary discipline, its primary focus should be to ensure that sound testing practices are followed throughout the entire factory. For example, developers should be writing unit tests for the code they develop and helping to incorporate those tests into an automated part of the software build process. User interface and interaction designers need to engage with test development to ensure that the user experience meets design objectives.
- **Automate everything.** Your customers don't care about your internal inefficiencies, manual processes, or friction points. What they care about is the software experience you deliver. Anything that gets in the way between you and what your customers want is moving you in the wrong direction. Manual processes are not only slow, fragile, and error-prone, they also represent opportunity cost — effort that could have been spent on creating new value. Of course, not everything can be automated, but everything that can should be. Automation will require up-front investment with downstream payoffs. You will need to be patient with your investments and persistent with your efforts.
- **Prioritize architecture.** In the constant rush to deliver the next set of features or functionality, it's easy to lose sight of the longer-term evolution of your software and systems. A constant short-term focus will ultimately result in a build-up of technical debt — an accumulation of technical shortcomings that over time creates a tangled web of complexity and fragility. Left to build too long, technical debt is very expensive to pay down and will limit your forward progress while you do. Instead, you must balance both near-term deliverables and long-term evolution so that your factory never becomes obsolete or mired in its own legacy.



Cognizant Technology Solutions on modern testing approaches

Guljeet Nagpaul, Director, Digital Assurance, QE&A at Cognizant Technology Solutions, offers the following remarks on Customer Experience (CX) and Quality Assurance (QA) testing in today's business environment.

Customer sentiment analysis is performed using specialized analytics systems that mine data from customer feedback channels (such as social media, app store and review sites, etc.) to arrive at sentiment scores.

These provide insight into what customers love and hate, what works or doesn't work, and what they want (or don't), as well as how a brand compares with its competitors. All of this information can be leveraged for better quality assurance — such as new feature verification, product updates and fixes, new test cases and A/B testing scenarios.

Crowd testing that involves independent end-user testers is another popular technique that can provide similar QA insight.

We applied all of these techniques while working with a large U.S. apparel manufacturer to perform CX testing on its fitness tracking wearable product. Huge amounts of data from the company's social media channels (such as Twitter) were mined using Hadoop-based big data analytics systems. This analytics data informed the development of more than 70-odd product feature improvements and new test scenarios that were previously not anticipated.

Similarly, a large pizza chain suffered from poor user experience on its mobile app (i.e., a score of less than 2 on a scale of 5). It worked with us on customer sentiment testing to rectify the situation. The company utilized crowd testing on its app as well as analysis of app store reviews to identify and address critical issues that resulted in a doubling of sentiment scores (to above 4).

In conclusion, CX testing represents a key departure from traditional testing techniques. Today's digital organizations need to understand and appreciate that customer experience testing is — and will always be — a continuous journey. As the digital world evolves with the advent of new technologies, channels, devices and options, so do the principles of customer experience assurance.

4

The Agile Journey

The path from idea to business outcome — fueled by the need to innovate for competitive advantage — is now the critical path at the heart of your business. The software factory that makes the idea real must produce at a rate that allows you to deliver value to your customers at the speed they want to consume it; any slower means missing customer demand. Waterfall development — with its monolithic projects, long lead-times, and static requirements — falls down on delivering both speed and value. In stark contrast, agile methodologies decrease cycle times and increase delivered value through the continual iteration of small teams freed from the chains of bureaucratic development processes. Because that iteration continually revisits what you are building in the context of what your customer wants, you increase speed by reducing rework, and increase value by intensifying engagement around the idea.

Agile has taken the software world by storm over the last couple of decades. Although the concept was originally introduced in the 1970's, agile methodology began to spread rapidly following the publication of the Manifesto for Agile Software Development in 2001 (often referred to colloquially as the agile manifesto). Agile is essentially an alternative to traditional waterfall or linear development processes. It emphasizes small teams or “scrums” that work in two-week intervals or “sprints” to continually assess and reassess project status and direction. The end of each two-week period is designed to produce a potentially shippable increment of software.

Regular customer feedback is an essential component of the agile approach, and the delivery of software in short, frequent cycles allows ongoing “inspect-and-adapt” sessions that power continuous improvement. Such introspection is a powerful tool to ensure that what is being built delivers the value that customers need, and that development never gets out of touch with the outside world. Because agile relies on highly cross-functional and collaborative teams, customer feedback and insights are shared broadly and inform everyone, rather than having customer feedback managed through dedicated channels. This allows agile organizations to make faster and better-informed decisions responding to their customers and prioritizing their needs. Such rapid responsiveness includes the ability for teams to self-organize and adjust themselves to address new or emerging needs.



Responsiveness and rapid adaptability relies on feedback and the open flow of information. Transparency and sharing of data isn't merely a question of etiquette; it is a necessity in an agile organization to ensure that decisions and priorities made and evaluated on an ongoing basis at a team level are made on the best information available. The distributed nature of agile-based decision making requires information to be freely distributed as well, and must flow both horizontally as well as vertically. Such transparency of information turns the notion of “information of power” on its head: information is indeed a powerful force, but only if it is freely shared.

Agile may seem like somewhat of an ad-hoc approach to the uninitiated. On the contrary, it uses highly disciplined project management and control of the flow of work to minimize waste and maximize total effectiveness. Agile takes the structure that a robust process provides and focuses it on documenting your development process as it unfolds. As some agile proponents say, it's like having a process that does its job and then gets out of your way. This can have significant positive cultural implications for your developer community. In contrast to waterfall methodology, which typically uses a top-down, command-and-control approach, agile pushes responsibility down to your teams and enables them to own their work. Agile's focus on small teams working on small batches of work also allows limiting the amount of work in progress (WIP) to avoid the inefficiency of time-slicing work across too many work items or projects. Disciplined limiting of WIP increases both team and individual productivity and is essential for agile effectiveness.

Creativity feeds on the diverse and frequent iterations that agile offers. If done well, agile makes value creation the central activity — rather than adherence to falsely-precise schedules with an emphasis on process over outcomes. Speed and innovation happen naturally when team members are freed to collaborate and co-create in agile “sprints.” And by removing rigid processes and management, it brings back some of the humanity to software development as a craft and a worthy endeavor. Teams that truly adopt agile develop tighter bonds and increased trust. This leads to higher team performance, lower costs, and faster delivery.

DELIVERING CUSTOMER VALUE THROUGH AGILE

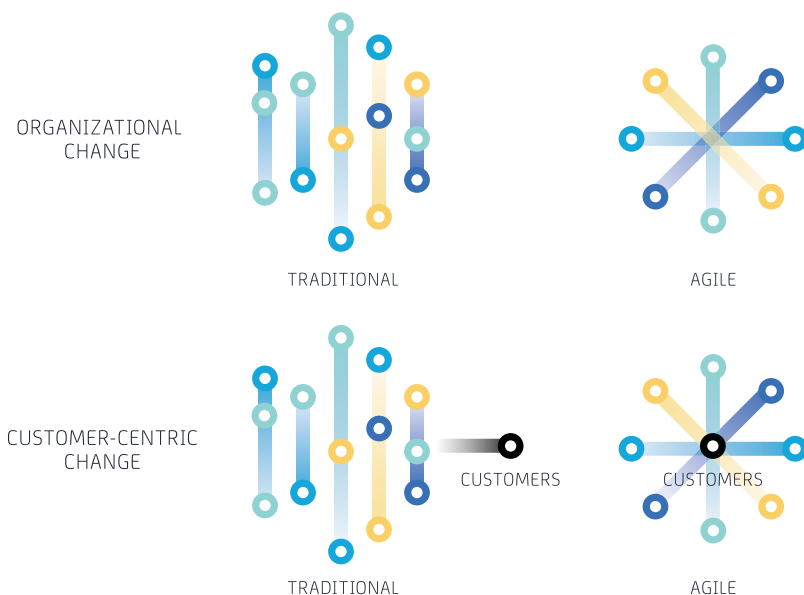


Figure 4-1: Moving to agile requires a transformation from functional silos to small, multi-disciplinary teams that iterate frequently and actively engage customers in the development process. Agile has the advantage of delivering value quickly and aligning to what customers want versus traditional slow-moving “waterfall” development methods that struggle to keep up with customer needs and rapid market changes.

The Road to Agile

Adopting agile is like installing state-of-the-art assembly lines in your software factory. When combined with increased automation and streamlining of the “ops” part of your software factory, you start to have all the pieces required to achieve the delivery velocity you need today. The benefits of agile, however, are not limited to software development; they can scale across the enterprise. What started as a software development approach is now playing a central role reshaping how businesses operate — another example of the deep impact of digital transformation.

Recent research shows that agile is on the critical path to improving the business bottom line and customer experience:

- Agile companies have been found to grow revenue 37 percent faster and generate 30 percent higher profits.¹

- The Project Management Institute found that organizations that rely heavily on agile methods are 50 percent more likely to successfully complete their strategic initiatives.²
- A summary of research on project management methods found that agile approaches yielded 29 percent better cost, 91 percent better schedule, 97 percent better productivity, 50 percent better quality, 400 percent better satisfaction and 470 percent better ROI than the least effective traditional methods.³

The speed gained with agile improves companies' time-to-market advantage and allows them to generate new revenue sooner. Agile taps the voice of the customer early and often, ensuring that the development organization is building the right products and features and delivering with quality and predictability.



Regardless of the size of the company — enterprise or start-up — it's no longer enough to simply implement agile for software development. To improve the odds in today's disruptive environment, agile practices must be pervasive throughout the organization and adopted at full scale.

Scaling Agile

The case for scaling agile practices is rooted in software's new and expanding place in the value chain. If the ability to deliver innovative experiences is a central determinant of competitive differentiation, then it stands to reason that your entire software factory needs to be optimized to deliver on that strategy.

Agile is inherently about optimizing software development across a particular project or a series of related projects. With the number of development projects, releases, and resources dedicated to delivering customer value constantly growing, having a holistic view of all your development investments becomes a business imperative. A deep understanding of your software portfolio is critical to ensuring that your investments are strategically aligned and provide an appropriate return to the business. This is where scaling agile across your entire portfolio can drive greater transparency around your development investments, give you the opportunity to optimize those investments, and ensure that they align with your strategy. Scaling agile across your organization also allows you to identify gaps and include new requirements at a portfolio level to allow you to predictably evolve the brand experience. Depending on your appetite and need for large-scale change in your organizational culture, agile principles can be extended to functions outside of software development such as finance, marketing, and other areas to achieve similar benefits.

Scaled agile isn't simply a software development methodology. What started as a movement around software teams has matured into an enterprise-scale business approach spanning teams of teams in distributed locations across a spectrum of industries. At scale,

agile organizations need to be agile-informed in their organizational structure, processes, ceremonies, and culture. Importantly, agile is an operational model that enables companies to use business agility as a competitive advantage.

In short, agile at scale introduces organizations to a new way of working, thinking, and delivering value.

Roadblocks to Agile Adoption

The highest level of agile adoption requires organizing people around the creation of value. That requires groups to break down internal silos, form value-focused teams, and connect agility across the entire organization. By aligning people and teams around customer value and taking a disciplined approach to managing change, organizations can become responsive and better able to process innovative ideas and deliver them to customers. Deep adoption of agile requires changing established culture and will challenge even the most evolved agile supporter.

Organizational silos

Disparate, silo-oriented teams will require leaders representing multiple domains to advocate and drive agile adoption. A successful agile effort not only requires real buy-in from executive management, but will also demand agile supporters in leadership across business, technology, organizational, and personnel groups. An empowered dream team of leaders from diverse groups must keep agile on track and enable it to scale throughout the entire organization. In addition, champions at all levels within the organization are critical when it comes to driving agile adoption in a grass-roots fashion. Agile is not a top-down methodology, and successful agile adoption can't be either.

Resistance to change

Many might balk at making change an accepted part of the development process. Waterfall development locks down scope requirements so that schedule and cost can be planned and it minimizes feedback during development. This can result not only in subpar code quality but also in delivering features and functionality to customers that they don't value. Agile recognizes that scope is always a variable, and it welcomes feedback as a critical and inextricable part of the planning process that continues through to delivery. In agile, change is welcome as an integral part of optimizing outcomes.

Changing team dynamics

As agile efforts scale, more teams need to be added. The underlying power of agile is in the collective power of small, individual teams. Successfully scaling agile requires more teams, not bigger teams. The teams themselves must be empowered, self-organizing, and self-managing. Even as the number of teams grows, each team must be responsible for delivering a set of well-defined, customer-focused results. And as the number of teams

increases, leaders must layer in practices that maintain transparency while coordinating the work between and among teams. Scaling agile requires a level of resolve, focus, and leader maturity to engage and direct what are essentially self-directed, but linked, teams.

The need for ongoing evangelization

Agile must not turn into a one-off project; it's a new way of working — and even thinking — within an organization. To have all aspects of agile successfully adopted across an entire enterprise, agile advocates have to keep touting agile's value and proving its benefit at every stage across every team. As more teams join the effort, leaders will be able to create mindsets and behaviors that drive ongoing improvements. The agile journey requires a belief in the fundamental value of agile principles as well as an understanding that it's a journey that never ends.



Getting Started

Don't get fooled into thinking that loosening up some parts of your development cycle for more iteration is the same as “doing agile.” Agile is an end-to-end system, which is why it is able to effectively drive systemic change. By the same token, agile is not a religion. It's a system for producing outcomes that create value for your customers. Your system is not the driver, your customer is. And don't get locked into a fixed formula in your move to agile. Use agile principles in implementing your agile program, and adjust as you go based on your outcomes and the insights you gain on your journey.

Scaling agile⁴ — especially in large enterprise companies — isn't just about adding more agile teams. Agile approaches may seem conceptually simple, but scaling agile isn't easy. It requires coordination, a willingness to learn, lots of patient practice, and a commitment to the tenets of collaboration, visibility, and continuous improvement.

1. Focus on velocity.

This first step on your agile journey is to focus on improving velocity; it's the heart of your agile transformation. Get your software and technology teams finishing their deliverables — with quality — as quickly as possible by building stable agile teams and delivery groups, and by reorienting people to thinking about delivering on shared outcomes. Improvements only matter if the organization delivers value all the way to the customer more effectively. Consider using metrics such as Net Promoter Score (NPS) to measure value, and be sure to track the number of releases delivered to customers per year to measure velocity. As you start to build velocity, you will be able to free up additional development resources and create a virtuous cycle of improved efficiency and value creation.

2. Implement agile planning and work sequencing.

Choose the most valuable, smallest pieces of work to deliver rapidly to your customers. Think carefully about planning the sequence of work and being able to deploy work more consistently. Limit the amount of work in progress across

development and operations teams, and account and buffer for unplanned work. Include design, development, operations, architecture, security, compliance, analytics, and other relevant stakeholders in your delivery groups and in planning sessions. Provide all delivery teams visibility into each other's WiP using appropriate agile tools, and share feedback openly from stakeholder and customer sessions.

3. Apply agile portfolio management.

Evaluate the end-to-end cycle in delivering new value: from the moment you think about doing something; to the decision to proceed; to funding and allocating the right teams; to moving it through the development organization; to deploying it to customers and implementing it throughout the business. Based on value, you must regularly decide which projects to start, keep going, or stop. Determining value requires a tight connection between the implementation organization and the business need. Eliminate the expectation of exact requirements tied to a precise budget delivering a perfect solution to help you strike the right balance between strategy and tactics.

4. Optimize investments on an ongoing basis.

Dynamically and intentionally reduce or amplify your investments through quarterly steering and release planning. On the other end, make sure you're getting the full value out of what you've invested — that you're delivering what your business needs and what your customers want, when, where, and how they want it. Optimize what you build for the greatest impact. Support this by emphasizing mindsets and skills of disciplined opportunity exploration (using methods such as Enterprise Lean Startup⁵) and by limiting organizational WiP. At the center of limiting WiP is a reduction in batch sizes enabled by short iterations, each of which is able to deliver a small portion of a larger segment of value.

5. Sense and respond with business agility.

Articulate shared outcomes, demonstrate high-trust leadership, and adopt a pragmatic, problem-solving mindset throughout your business. Cultivate organizational health by clearly expressing these same goals, establishing trust-based teams through regular and open communication, and by collaborating across as well as up and down flatter management structures. Create what may at first feel like threatening levels of transparency, and delegate decision-making all the way down.

Establish structural flexibility with adaptable funding models, and understand how to flow work to shape your business on demand. Become able to sense new opportunities in the digital age by welcoming — in fact, expecting — innovation to come from everyone by giving every single employee a charter for disciplined exploration.

Making the move to adopt agile practices is difficult and requires tenacity as well as sensitivity to the cultural aspect of the transformation. But by embracing a different way of thinking and operating to accomplish the distinct, high-impact activities outlined, you will be able to achieve results that may have previously seemed unattainable.



Partner Perspective

PayPal takes a big-bang approach to agile to increase speed to market⁶

When Kirsten Wolberg joined PayPal a few years ago as a new VP of Technology, everyone she spoke with in the development organization told her the same thing: it was really hard to get work done. PayPal was relying on waterfall development methodologies and, because their workforce was so specialized, there were 85 bottlenecks within the development process.

One problem was that the development teams were focused on very large-scale development projects that would take years to get out the door. Huge product specification documents would be thrown over the wall to development teams. By the time the product was released, there was often a complete disconnect between what customers wanted and what the technology teams delivered.

Additionally, PayPal was operating with a “project mindset,” rather than a “product mindset.” If a technology leader had a product idea to develop, she or he would assemble developers from across the organization. After the release was out the door, those developers would be assigned to the next new project. As a result, products were essentially orphaned after they were rolled out; there was no one who understood the code and maintained consistent ownership of the product and whether it was meeting customer expectations over time.

PayPal began experimenting with some pilot agile projects, introducing the concepts of minimum viable product and dedicated scrum teams. They began seeing some benefits right away. “With teams in place that fundamentally own the products in the marketplace, they took responsibility for the customer experience. They care about the quality of that product. They care about what the next release is going to look like, and how it’s going to meet customer needs. That mindset shift has resulted in much better experience for our customers, and a better experience for our teams, because they have much greater pride in the work they’re doing,” said Wolberg.

With these initial successes in hand, Wolberg pushed for a “big bang” approach to scale agile quickly to the rest of the development organization. She realized that because development teams are so dependent on each other, PayPal wouldn’t see the full benefits of agile until they had all teams adopting the methodology. Additionally, Wolberg feared that because everyone would feel the pain involved with change but not see the benefits in the short-term, executive support for agile might wane if they didn’t move quickly enough.

After seven months of planning and training, PayPal moved from having 20 percent of its development organization following agile approaches to its entire technology and

product organization of people adopting agile, scrumming in two-week sprints. The results speak for themselves: in the 18 months prior to its full-scale adoption of agile, PayPal had rolled out three new products. In the six months after full-scale adoption, it had rolled out 58 new products and features. “By forcing the entire team — 3,500 people across four major development centers and 40 global business offices — to move on the exact same day to a new way of working, the entire organization began to see the benefits immediately,” Wolberg said.

5

APIs — Your Digital Nervous System

Less than a decade ago, everything was about the web: businesses were scrambling to build websites to provide portals to their systems and services. This initial step towards digital engagement focused on providing access to existing businesses in a new digital medium. The explosive growth of Internet-connected mobile devices has driven the next phase of this digital evolution in the form of multi-channel applications: entirely new software experiences delivered to an array of devices ranging from PCs to watches. While early websites aimed to extend existing relationships with the business using technology, the explosion of mobile apps allowed — and in fact demanded — fully complete, customer-centric experiences. Digital engagement and interaction with the business any time, any place, and on any device has become a basic expectation. Digital engagement is no longer a mere enhancement; it is now a presumed primary form of interaction with any business.

Augmenting a business with a website is one thing; delivering new experiences on a broad range of devices to engage deeply with a business through digital channels is another. The flow of data between the business and its partners, employees, and customers quickly becomes complex and threatens to be unmanageable. Meeting the challenge requires a unifying approach — some common way to achieve coordinated communication between an increasingly complex matrix of connected devices, businesses, customers, and experiences. The organizing force that has emerged is the Application Programming Interface, or API.

Building a Connected Business Using APIs

There's nothing breathtakingly new about APIs — they have been around as long as computers have. At their essence, APIs are a way for one piece of software to communicate with another that is requesting the execution of some predefined function or service. A simple example is an application obtaining the current time of day for a specified time zone. That functionality can be developed once, and then made available via an API. Any application can then obtain the current time through the same time-of-day API. The most common use of APIs has been to enable the development of applications on a device-specific platform — e.g., a Windows PC or an iOS mobile device. The APIs used for building these types of applications are unique to the device platform; Windows APIs used to build PC applications are very different than iOS APIs used to build applications for iPhones, iPads, and Apple Watches. Furthermore, these platform-specific APIs are part of the device platform itself; the apps, and all of the software that makes up the APIs they use, all run on the device.

Web APIs change all of this by making it possible to access APIs via the Internet; they do this by using platform-agnostic web-based protocols for the program interfaces. This makes it possible for any application on any platform to access the same functionality through a uniform, universal interface. It doesn't matter if the app is running on Windows or a Mac or an iPhone — access to the web API is the same. The execution of the API request may happen on a server halfway around the world, but from the application's perspective, the API could be running on the same device as the app. Web APIs remove distance as a barrier and enable real-time, worldwide access to an ever-expanding catalog of functionality. Using the simple example of getting the current time implemented as a web API, the API request and the information that the request returns (the current time) will be the same whether the request is made from a PC or iPhone or any other platform. Neither the physical location of the request nor the type of platform running the time API will have an impact on the API's use.

These two factors — easy access to distributed computing resources, and interface uniformity — are the basic drivers behind the tremendous value APIs provide. Any app on any device with an Internet connection can access the same vast catalogue of available web APIs. And millions of servers connected to the Internet make it possible to quickly turn an idea into a valuable API-based service available to application developers. For example, a business can build APIs to enable customers to get inventory information, place orders, and get order status. Customers can then use those APIs in their applications

to automate and streamline purchasing and order tracking with that business partner. Everything from user identity management to social integration to credit card transactions to location services — and so much more — are readily available via web APIs.

The universal nature of web APIs has also contributed to their rapid adoption. Unlike platform-specific APIs that require upgrades to the platform itself in order for applications to take advantage of enhancements and improvements, new functionality is available to all applications as soon as the changes to a web API are deployed. Furthermore, the platform-agnostic nature of web APIs makes it easy to combine APIs from many sources to create unique new applications, while minimizing the need for reinvention and duplication of efforts.

APIs – YOUR DIGITAL NERVOUS SYSTEM

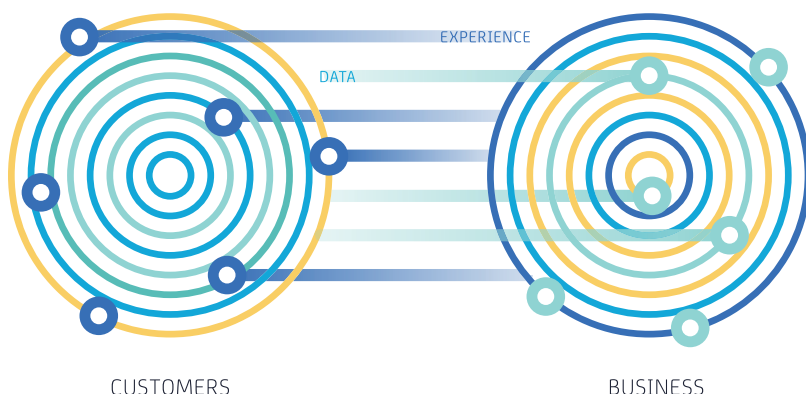


Figure 5-1: APIs have long been a way for one piece of software to communicate with another. *Web* APIs enable real-time access to an ever-expanding catalogue of functionality that is platform-agnostic. They are used to deliver rich experiences to customers by providing valuable data as well as delivering data back to the business.



Because APIs are such a powerful tool for sharing data and services, they are driving a complete reengineering of the software supply chain. By programmatically providing access to a business's information and services, APIs dramatically accelerate application development speed. As the L'Oréal example later in this chapter helps illustrate, APIs have become essential building blocks of digital transformation and the application economy. The rapid build-out of cloud-connected, feature-rich mobile applications and services would simply not have been possible without APIs.

There are currently 14,667 public APIs¹ available, powering everything from financial transactions to location services to weather forecasts that deliver information and services to our laptops, phones, watches, and cars. Titans of the Internet age such as Google, Amazon, Netflix, and Uber have built their businesses on the power of APIs. For example, Netflix disrupted and redefined the media industry with an API-based approach that enabled the digital delivery of a large library of content to a wide array of devices and a consistent, high-quality experience. Not only are APIs driving new functionality, but they are also contributing substantially to revenue. A *Harvard Business Review* article states that Expedia generates 90% of its revenue via its APIs, while eBay clocks in with 60%.²

The API Advantage

The value of APIs is not limited to pure technology players alone. The app economy opens the playing field to all businesses who can leverage APIs to fuel a wide variety of initiatives across a diverse set of industries. The leaders of tomorrow will be companies that embrace web APIs to create superior new integrated solutions and experiences for their customers. Research from Freeform Dynamics, an independent IT research and analysis firm, stresses this broad impact and reach: enterprises with advanced API implementations are 2.6 times more likely to have faster delivery of revenue-enhancing apps and 2.3 times more likely to have lower IT-related costs and risks.³ A sound API strategy serves a host of possible business objectives — from acceleration of app development to driving consistent user experiences to accessing new channels.

Acceleration of application development

A primary benefit of APIs is the sheer speed at which new applications and features can be built and delivered. This benefit extends beyond your own APIs; a wealth of third-party APIs can help you build your software experiences without having to reinvent the wheel. Need a map in your app? You can use the Google Maps API. Facebook social integration? No problem. Credit card processing? There are services with readily available APIs to handle that. Twitter, YouTube, Flickr, Spotify and many more all offer web APIs to enrich the user experiences you deliver with social and online media.

Even if you're not a programmer, using APIs to perform basic Facebook integration is a snap, as Figure 5-2 makes clear.

The speed at which APIs can create new value gives your organization the ability to respond quickly to market changes and new customer demand.

Delivery of quality multi-channel experiences

APIs give your organization the ability to connect systems and share data to deliver a consistently high-quality customer experience across multiple channels. Although the target

device platforms themselves may be very different, the underlying APIs that feed them with data and services to power the experiences you deliver can be engineered to be uniform and consistent. This API-level consistency helps to ensure consistency and quality when it comes to the user experience. The device-agnostic nature of web APIs is what makes them such a powerful tool for aggregating and composing functionality in a uniform way even across a very diverse set of devices. For example, suppose you need to display images of your products on apps that run on phones, tablets, PCs, and game consoles. One option would be to implement a separate API for each device type. This approach would create needless complexity that would increase the potential for errors and the possibility that product images would not be consistent across devices. Instead, you can implement a single API to provide the appropriate image for each device. This approach reduces overall complexity and the chance of errors or unintended inconsistencies, and helps ensure a seamless, high-quality multi-channel user experience.

Graph API Call

Let's make an API call to publish a message. Add the code into the response function of the **FB.login** call you added above:

```
FB.login(function(){  
  // Note: The call will only work if you accept the permission request  
  FB.api('/me/feed', 'post', {message: 'Hello, world!'});  
}, {scope: 'publish_actions'});
```

Try the script. A status message will be posted to your timeline:



Matthew Johnston
4 seconds ago via Reli 

Hello, world!

Like · Comment · Promote · Share

Figure 5-2: With APIs, you can program without extensive programming knowledge.

Access to new business opportunities

APIs are a powerful tool for creating new business opportunities. They allow you to expose valuable data and information that you already have in order to build new products and services, or to enable a partner-based ecosystem around your own business. APIs allow you

to take a database — or to aggregate data from multiple data sources — and expose the valuable information they contain programmatically as live services. Such API services can form the basis of broadening access to your business and the value that it provides. For example, Weather Underground offers a tiered pricing structure for its APIs and targets them directly to developers looking to enhance their apps with weather-based information. And Sonos offers third-party developers access to their APIs in order to bring more value to their devices by enabling an app ecosystem through their API platform.



Partner Perspective

Powering L'Oreal's digital transformation through APIs⁴

L'Oreal began embracing web APIs two years ago to accelerate its digital transformation. The company needed technology designed to be sharable, flexible, nimble and scalable — enabling L'Oreal to create the personalized digital services today's connected consumers expect.

"We wanted to up our digital game at L'Oreal. The L'Oreal philosophy is to be very, very forward looking. APIs are really the way of doing business for us right now," comments Susannah Greenberg, Vice President, Technology at L'Oreal.

At L'Oreal, APIs improve partner communication and development because they enable backend systems to more quickly and efficiently share pricing data and product availability across vendors, sites, L'Oreal brands, and platforms.

APIs provide agility and cost advantages, too. The company can tap into partners' merchandising data using APIs. Having an API gateway and well-defined abstraction of API services means L'Oreal can switch between third-party service providers (such as an email marketing provider) or adopt new capabilities faster. As a result, APIs have enabled L'Oreal to support a fivefold increase in the number of endpoints with partners and vendors without expanding its internal teams.

Today, nearly a dozen APIs now power some of L'Oreal's key consumer services, enabling a variety of features. They include front-and-center beauty apps, such as Makeup Genius, which uses facial recognition technology to let shoppers virtually try on 420,000 products from lipstick to eyeliner. By combining their internally developed APIs with third-party APIs, L'Oreal has been able to create unique and compelling digital experiences that deepen the company's engagement with its customers.

API Challenges

APIs are an essential ingredient in your digital transformation and they can rapidly expand the reach and potential of your business. As with any major software initiative, however, there are a number of important areas to address as you plan and implement your API strategy.

Design

A common mistake with APIs is thinking: “We have all this data we’re going to expose with APIs.” An ad-hoc approach to API creation will lead to wasted effort and a jumble of APIs that don’t get used or can’t be used efficiently. The APIs you build must focus on enabling new features and scenarios. The design of your APIs must also consider the needs of developers — the users of APIs. APIs that are difficult to use or error-prone can cause delays, needless effort, reduced adoption, and compromised application quality. APIs should make the job of developers as easy as possible by hiding the maximum amount of work and complexity behind the API implementation. As a simple example, consider an API that returns the results of a database query in a product catalog. Suppose that the query may produce a number of unhelpful redundant results — identical copies of the same product in the list. One way to design the API would be to leave those redundant results in place for the API user — the developer — to deal with. A better API design removes the duplicate items in the list before returning the results. Doing so reduces the amount of work every developer using the API must do, and decreases the chances of unintended consequences or side effects.

Integration

The existence of multiple API methodologies and data formats can also cause some headaches. Depending on the requirements of your API users, you will need to develop APIs that support one or perhaps even both of the two prevalent data exchange formats (JSON, short for *JavaScript Object Notation*, or XML, which stands for *eXtensible Markup Language*). Although REST (*Representational State Transfer*) is the most common method of accessing APIs, other methods may be more appropriate based on the specific use. For example, MQTT (*Message Queuing Telemetry Transport*) is a common interface for IoT and some mobile applications, and SOAP (*Simple Object Access Protocol*) also addresses specific use cases. In addition to being thoughtful about API protocols in the design of your APIs, your development team will also need to be adept at using APIs external to your organization, since they will be using a variety of API protocols that may be different than your own.

Security

APIs are not inherently secure. By their very nature, APIs provide access to often highly sensitive data at the core of your business and magnify the dangers of any lapse in security. It is critical to ensure that your APIs don’t compromise your sensitive data, and that

only the right set of users, apps, and devices can access your APIs based on the appropriate controls. You must integrate security deeply into your APIs with rigorous design and security measures such as encryption, robust authentication, and risk management. There are many examples of security lapses; some are as basic as a username and password hard-coded — embedded directly — into the API itself. In another instance, a non-secure API was used to download all the usernames associated with an online service; these were then used to attack other popular services. You don't want to make the news this way!

Scalability and performance

Consistently high performance of your APIs is critical to delivering a great user experience. We've all encountered — and been frustrated by — crashes, hangs, or lack of responsiveness trying to access an app or online service. The cause is often an API failing to perform. Popularity of your service and the APIs that power it is a good thing, but you have to ensure it can scale. You must architect your APIs and all of the software and systems supporting them so that anticipated usage won't be overwhelming when traffic increases. Your APIs should also be designed to gracefully handle variable network speeds and latency. For example, your API can be designed to provide lower-resolution and more space-efficient images as needed to decrease the amount of network traffic — and time — required for the user to see images on their display. Or your API can return a batch of data in a specified range to decrease unnecessary back-and-forth between the app and the API that can be very time-consuming when dealing with poor network responsiveness.

Discoverability

It can be difficult to find the right API for the right task, and your APIs will be no exception. APIs are not a “build-it-and-they-will-come” technology. If you want people — third-party developers, partners, and your own developers — to use your APIs, you need to get the word out and document how to use them. This should include sample code that can serve as a helpful starting point for developers learning to use your APIs.

Lifecycle management

Creating a new API is just the beginning of a lifecycle through which the API will evolve, mature, and ultimately reach the end of its usefulness. Building a new API is a long-term commitment, and you should strive to have the fewest number of APIs possible. It is all too easy to build a proliferation of APIs, causing a confusing jumble of functionality and a support and maintenance nightmare. You must include the evolution of your APIs as part of your design process to ensure that you build in both forward and backward capability. Future versions of your APIs must continue to support apps using previous versions, and you have to leave enough flexibility in your initial designs to allow API evolution and the addition of new capabilities to future versions of your APIs. As your organization's API savvy and use grows, you need a system in place in order to track API versions, gracefully sunset old or outdated APIs, and ensure the latest API versions are adopted as quickly as possible. A forgotten older API could be an unintended backdoor into your systems, and

you also have to be mindful to maintain backward compatibility as needed to ensure that your partners' and your own older applications continue to operate.



Getting Started

APIs are an essential tool for delivering new value to customers and partners quickly and at scale. A few critical considerations will help your efforts deliver on their full potential:

- **Invest based on value.** Just because you can build an API doesn't mean that you should. Avoid the temptation to build new APIs simply in response to their popularity; choose your API investments based on their value to your customers and your business. Carefully evaluate the use cases for new APIs — or enhancements to existing ones — to ensure that they are clear, compelling, and provide lasting value. Any new API you develop should have at least one initial user to ensure that the API is fulfilling a concrete and well-defined need.
- **Plan and build for the long run.** It's important for you to take the long view in building out your API program. APIs that you build and deploy today may be in use for many years. Architect your APIs carefully to be able to adapt to the changing needs of your business. Maintain the smallest set of APIs possible to minimize development overhead and needless complexity, and keep in mind that every API you add to your portfolio has an ongoing cost. APIs are a foundational technology; you must ensure that your APIs are designed and built with sound architectural principles at a high level of execution quality. Any deficiencies in the design and implementation of your APIs will be magnified as their use increases; make sure that you quickly address any shortcomings and are constantly improving the quality of your API technology.
- **Focus on your developers.** Developers are the target audience of your APIs. Just as you must consider the value that an API delivers to customers, you must also consider the developer experience with your APIs. Your goal must include maximizing developer productivity. This includes high-quality design and execution of your APIs and high-quality sample code and documentation to accelerate adoption and increase the effectiveness of your APIs. You must be ready to support your APIs with problem reporting, tracking, and resolution. Your APIs should do as much work as possible to remove as much complexity and overhead from the developer as possible. Building the needs of the developer into your API program will help ensure both development velocity and product quality.
- **Scalability, security, and performance are essential.** A well-designed, well-supported API portfolio is just part of the API picture. Your APIs must also deliver consistently secure, reliable, high performance operation. Security must be a core feature of every API you build to ensure that you never compromise your customers' trust. You must ensure that only authorized users have access to your APIs, that you protect sensitive data and privacy, and that you respond quickly to new threats. Performance and reliability are also at the heart of a high quality user experience. You must design and implement your APIs to provide consistently high performance and to be able to quickly scale to accommodate use.



Partner Perspective

PwC on containerization

Munawar Lakdawala, Principal, CIO Advisory Services for PwC, shares thoughts on the opportunities provided by containerization technology.

Organizations have used technology as a key business enabler to grow and expand rapidly for many years. However, with such growth comes ever increasing IT costs and businesses are constantly striving to contain these costs while still reducing the time to market for new features or enhancements of business applications. As a result, IT managers are constantly seeking to innovate and enhance their processes to drive efficiency in rapidly delivering solutions to the business while responding to cost pressures.

Managing IT costs while delivering efficiency has become the standard model and DevOps has been front and center in that discussion in today's IT organization. One of the technology enablers for DevOps is software container technology, which has emerged as the disruptor in the traditional application development infrastructure landscape while generating excitement with IT managers. This technology has proven to facilitate the creation of a self-sufficient, lightweight, and portable application environment while abstracting the environment's dependencies and configurations.

Developers and testers can provision production-like instances using containers without any delays associated with traditional virtual machine infrastructure or time-consuming platform build cycles. These container technologies abstract the necessary application components without the need for a full stack operating system relying on heavy and redundant infrastructure. As such, container technologies are lean and far more cost effective when compared to conventional virtualization technologies. A typical IT organization can now leverage container technology to address multiple challenges simultaneously, such as:

- Painful developer onboarding process
- Lengthy environment build cycles
- Redundant and inefficient infrastructure
- Lack of innovative IT solutions
- Inconsistent provisioned environments for developers and testers
- Environment-related defects due to varying configurations
- Reduced scalability due to heavy infrastructure footprint
- Lack of speed and agility in responding to business needs

Addressing these challenges simultaneously is what IT managers and business executives demand. However, incremental results can be realized quickly and can begin to address these challenges with careful planning and adoption of container technology. This further enables IT to increase its DevOps maturity at a sustainable pace. In order to get started with the adoption of container technology, it is important to identify the most important use cases — be it developer, environment or modularizing the delivery pipeline in the target environment.

Another important aspect to consider is the underlying platform on which the containers are expected to run and any additional infrastructure needed to host the container registry and images. This can be managed by implementing a governance model that allows for managing multiple versions of different containers via a robust configuration management solution. The result is an accelerated adoption of containers within the developers community of your IT and ultimately DevOps as a whole.

Container technology has begun to transform the way IT delivers solutions. As a repeatable, scalable, and standardized solution, container technology has allowed IT to cut costs and reduce the time to market while improving the quality of developed solutions for the business. As an integral part of the DevOps landscape, container technology enables the IT organization to deliver against their SLAs in an innovative manner.

6

Integrating Security to Fuel Your Transformation

Today's software-driven world is dominated by users looking for new value and fresh experiences. Delivering those experiences at speed and scale means that at the heart of every business today is an “experience supply chain” that requires predictability and efficiency to function. Any disruption of your ability to deliver the experiences your customers demand can damage your business and your brand. This is why securing those digital channels has become a critical business process. The difference, however, between physical supply chain security and the digital kind is that security not only impacts the delivery of the experience, but it can deeply impact the experience itself. Digital security then has two main objectives. First, it must protect the channels between you and your customers from external threats such as hackers, malware, viruses, and denial-of-service attacks. Second, it has to improve the user experience itself by reducing friction and establishing trust by using advanced approaches such as security analytics and automated identity governance verifying that users are who they say they are. Today, fostering ongoing security

innovation and incorporating leading-edge security technology is essential for building and delivering great software experiences.

The Security Imperative

Ensuring security across all aspects of the software-driven enterprise has become increasingly complex in the face of rapid technological change. The news is replete with stories about refrigerators sending malicious emails, sophisticated cyber fraud, and cars being hacked while being driven. The challenges will only continue to grow as the number of Internet-connected devices and apps accelerates. It's predicted that there will be 200 billion connected "things" in the world by 2018¹. All of them will be communicating through software, and all of them will need to be secured to function safely and reliably.

Not that long ago, we just had a few passwords to remember; now we have to think about managing our identity and access across a myriad of cloud-connected apps, services, and mobile devices. The interconnected nature of today's digital ecosystem amplifies the risks. A hacked Twitter account can create immediate reputation damage. And while signing into multiple services with a single social media account is convenient, a compromised account will also compromise every service to which it was linked. Leaked passwords, credit card information, or other personal data can wreak havoc.

The critical role of security in our lives will only increase as the Internet of Things (IoT) introduces new types of vulnerabilities and further raises the stakes. This is because IoT is far more than a set of devices communicating directly with some servers — it is a complex, multilayered mesh that includes sensors, actuators, hubs, data repositories, applications, and messaging frameworks that together make up the IoT ecosystem. Each node in the ecosystem has a unique relationship within it, requiring independent, coordinated interaction without human intervention. Understanding the security implications of all of the relationships within an IoT system is critical to preventing a vulnerability that could turn into the next high-profile security breach on the news.

As the technology footprint of the enterprise expands, so too does the complexity and cost of securing it. At the same time, customer expectations around the safety and security of their digital experiences are increasing. While protecting customer data is the price of getting their business, customers today also demand a seamless, hassle-free experience. They expect you to make security robust and friction-free in a world where the underlying technical challenges are only intensifying. The cost of failure on either side of that challenging balance is extraordinarily high. Not only will users abandon your business if your security keeps them out, 10% percent² of them will abandon your brand forever if there's a security breach.

Today, security is much more than passwords and perimeter defenses. It's also about data science, anomaly detection, and predictive analytics — systems and processes that understand your customers' behavior and how your software operates and interacts. Security now also encompasses system self-awareness that can detect and potentially mitigate or stop data breaches, denial-of-service attacks, unauthorized access, and other security failures.



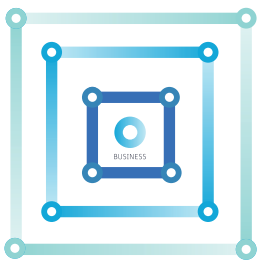
As you evolve your software factory, it's increasingly important to integrate security into the development process from the start. You cannot leave security to the end of the release cycle as a box to check off after everything else has been done. Security needs to be built into new products and services starting with the initial design. Security will have an impact on architecture, user interfaces, selection of APIs and services to use, technology frameworks, and even programming language choices. Security should be represented at the scrum team level throughout the entire development lifecycle, and a security expert is increasingly becoming a part of the scrum or integration team. Best practices should include coding standards that incorporate security considerations; every developer should have an understanding of the security implications in his or her area and how to write secure code. And new technologies that scan software code for overlooked security gaps or common coding mistakes can also be used to enhance security robustness.

Companies today face cyber attacks hourly or daily; the threats are never-ending and constantly changing. In the digital world, security is not someone else's job. It is everyone's job because security is the primary determinant of trust in a global digital society. Accountability for security increasingly ends with the Compliance Committee or Chairman of the Board rather than with a Chief Information Security Officer (CISO) because it is so foundational to any business today.

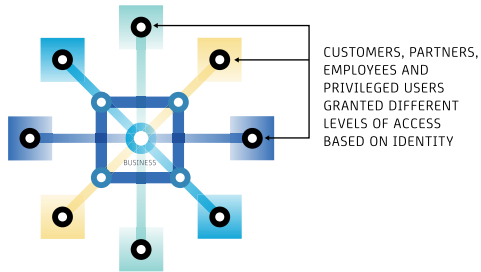
Security is an essential tool for controlling cost, preventing revenue loss, and improving the overall customer experience. For example, banks reportedly want to limit required user interaction to a small proportion of all ecommerce transactions today. Intervening more frequently than that becomes onerous for customers and can result in an increased number of transactions being abandoned with corresponding loss of transaction business. In card payments, for example, analytics allow you to understand a transaction in the context of what is normal for each individual cardholder. Sophisticated modeling techniques are used to assess risk in real-time by analyzing unique authentication data such as device type, geographical location, user behavior, and historical fraud data to separate genuine transactions from suspected fraud. Security analytics technology enables all of this to take place in the background, thus freeing most customers from the inconvenience of having to go through additional authentication steps.

While security technology can create new business efficiencies, the cost of a security failure can be massive. Home Depot's breach resulted in 56 million compromised cards and \$63 million in losses; there are 44 civil lawsuits. It cost an additional \$60 million to cover reissuing cards and related expenses³. The Target breach had 70 million stolen cards and cost the company \$252 million. The CEO resigned, the CIO was replaced, and they hired their first CISO⁴.

SECURITY MUST BE A BUSINESS ENABLER



SECURITY AS THE PERIMETER



IDENTITY AS THE PERIMETER

Figure 6-1: Traditional security paradigms based on passwords and perimeter defenses are giving way to approaches based on data science, anomaly detection and predictive analytics — systems that get to know your customers' behaviors. Ensuring strong security that doesn't get in the way of the customer experience is of prime importance and requires security be fully integrated into the development process from the start.

Identity Is the Perimeter

Digital transformation requires building new digital channels into your business, but doing so eliminates the classic enterprise boundary that had previously provided a degree of protection by having systems shielded from large-scale external access. In the transformed enterprise, the security boundary moves all the way to individuals accessing your company's data on the device of their choice, anywhere, anytime.

This dramatic shift of the enterprise boundary all the way out to the individual user requires rethinking security with the user's identity at the center, and it brings new challenges to consider:

- Low tolerance for inconvenience:** The explosion of mobility and the resulting expectation for fast, on-the-go access to apps and information has decreased user tolerance for intrusive or complex security processes and resulted in the “consumerization” of technology. Today's customers expect instant access; cumbersome, inconsistent registration and authentication processes can turn them away. Customers desire security with minimal disruption to the task at hand, whether it's checking a bank account balance or checking the status of an order. And the need for convenience isn't limited to consumer applications. Enterprise employees and partners are also demanding easy-to-use solutions to make it easy for them to access corporate information securely and efficiently.
- Proliferation of privileged accounts:** As businesses digitally transform themselves, access and control of tools and information is spreading throughout the entire

organization, outside the control of a single department or centralized function. While broadening the use of technology throughout the business is both a necessary and desirable outcome of transformation, it makes management of privileged accounts more difficult. We've come to think of privileged accounts as those owned by the IT types — network, server, and database administrators — but that is no longer the case. An array of third parties, vendors, and partners may also require privileged access to do their work. The person in marketing who runs your marketing automation tool also has a privileged account. If attackers gained access to his or her credentials, they'd have access to all of your customer data. The same goes for your online sales account management tool, cloud-based storage, and many other examples that may surprise you. More people have non-traditional privileged accounts than you might think, and these types of privileged credentials are involved in the majority of data breaches.

- **Threats are now everywhere:** The rapid expansion of security threats makes it impractical for a single department to protect the entire company. Protecting the company against security threats requires a much broader, multi-faceted approach. Security is a business challenge, not just a technology issue. After all, the cost of a successful external breach is approximately \$3.8 million on average⁵, and this does not include the cost of lingering business impact due to reputational damage. Ongoing education both within and outside of the technology organization will help drive understanding and awareness. A regular portfolio review approach focused on security across the entire business will help identify emerging threats, gaps, and mitigation strategies. Given the ever-changing and multiplying threat vectors, detecting vulnerabilities and breaches becomes as important as trying to prevent them. You should assume that malicious actors will find their way in one way or another, and you must be prepared to detect the intrusion and respond quickly to minimize any losses.



Getting Started

Tackling today's security challenges requires broad awareness and participation — everyone must be thinking about the implications of security. The security of “no” based solely on rigid, highly constrained systems is no longer viable. You need to adopt the security of “know” by building insights and dynamic responsiveness into your security strategy. These suggestions can get you started with integrating security as a critical enabler of your business:

- **Build in security as a killer feature.** You need to make security a core principal in all of your software development and deployment from the very beginning. Waiting until the end of the development process to bolt on security is a recipe for creating vulnerabilities and delivering a sub-par user experience. Agile organizations should have security practitioners as part of the delivery team, and security should be an intrinsic part of the planning process. Adopt a “defense-in-depth” approach by securing data and applications at every level of the technology stack. Design and test your applications and services for robust, secure operation while focusing

on delivering a friction-free customer experience. You will need to find the right balance between the ideal user experience and the required level of security protection.

- **Make identity your new perimeter.** The enterprise boundary is now the individual user accessing enterprise data via an application or service. Users are the new perimeter in the application economy, and this requires an identity-centric approach to security to ensure that users are who they digitally claim to be. Whether you build your own identity service, federate identity with trusted third-parties, or even enable some logins through social networks such as Facebook or LinkedIn, you must be sure that users are who they say they are and that the information and services they can access exactly matches their role.

Some data and services may need greater security than others. A user connecting anonymously or with a social login may have some level of access to public data, while other information might require a username and password. Augmenting this basic identity with advanced authentication protocols such as multi-factor authentication (a passcode texted to the user's phone after a login attempt, for example) can help ensure identity authenticity. Risk-based authentication can be used to dynamically adjust ease of use versus risk reduction. For example, a simple password may be sufficient to access balance information in an online banking scenario, but transferring funds may require additional identity verification in order to complete the transaction — an example of “step-up” authentication. Analytics-based security tools can add another layer of validation by combining different sources of data such as location, time of day, transactions, device being used for access, and other seemingly unrelated data to create a probabilistic determination of identity. Using this information, apps could determine whether to automatically allow access, or to request further proof via secret questions or similar unique identifying data.

- **Protect and monitor privileged accounts.** If attackers gain access to a privileged account, they have gained access to sensitive data and/or mission-critical systems and functionality. Insider attacks make heavy use of these accounts as well. You can take these two basic steps to minimize your privileged account exposure:
 - ***Grant the minimum amount of access needed for the shortest time possible.*** Providing more privilege than required is inviting trouble, and revoking access as soon as it is no longer required — perhaps even on a per-task basis — will keep the window of opportunity for abuse as small as possible.
 - ***Monitor privileged access at all times.*** Recognize attempts to escalate privileges as well as any behavioral changes from these accounts. If privileged access is compromised, account access histories will help you better understand what happened and why.

Delivering software experiences to your customers requires new pathways into your business that increase the risk of vulnerabilities. There is no longer a clean separation between the inside or outside of the enterprise and security can't be an add-on — it must be integrated throughout the technology stack of your business.



Partner Perspective

The Advisory Board Company sees security as a competitive advantage⁶

The free flow of information throughout the healthcare ecosystem is essential for effective health management and care efficiency. However, the security of that information is an inhibiting factor for many healthcare organizations that struggle with integrating provider and payer data into new and existing systems. And with data stored across numerous diverse systems, health care organizations find it difficult to use it to its full potential.

“The explosion of data in hospitals can provide insights in how to achieve greater efficiency and quality,” commented Ivan Sager, Director of Product Development at The Advisory Board Company, a global research, consulting and technology firm that is a trusted partner for 180,000 healthcare professionals in more than 3,900 hospitals and healthcare organizations around the world.

The Advisory Board has developed a number of products to help healthcare organizations tap into that data. With applications deployed into separate environments, they soon realized they needed a secure solution to integrate all the data while maintaining compliance with HIPAA and other regulations regarding patient privacy.

“As our products transformed into an integrated solution to solve broader healthcare challenges, it was obvious we needed a way to orchestrate these integrations in a secure, reliable, and flexible way,” commented Sager.

They developed a range of APIs to simplify the sharing of data, both internally and externally. Engineers designated as integration agents work with product teams and integration partners to author policies, publish services, and help evolve the APIs to deliver greater value to members.

Members are using these APIs to bring together and orchestrate their existing assets into one cohesive healthcare system providing both real-time and retrospective data. Members can now aggregate metrics for a specific physician — for example, the number of patients who fail to visit a primary care provider after their discharge — to build a continuum of care measures. “By improving how we manage our APIs, we can provide healthcare organizations with faster access to smarter data, which will help them improve the services they offer to patients,” commented Sager.

7

Measuring for Insights

In his 1982 book *Megatrends*, John Naisbitt stated: “We are drowning in information but starved for knowledge.” This is more relevant than ever in a software-driven world. Modern data gathering technologies and techniques have created a deluge of data, but very little of that data actually yields meaningful insights to guide decision-making. (Only 1% of all data collected from oil rigs, for example, is actually used to manage operations.¹) The combination of Internet connectivity and ever-increasing storage densities makes it tempting to collect and keep as much data as possible. But just because you can gather vast quantities of data doesn’t mean that the data will automatically yield insights. Effective use of data begins with some initial triage to attain a level of granularity and consistency where gaining insight is actually possible. This process can be helped by identifying the types of questions that would be valuable to answer, building the right data science models, and applying analytics tools to generate insights from your questions.

Exploiting the Data You Have

Generating insights from data is an essential capability to be able to put your customer at the center of your business and deliver an ever-improving experience. Your ability to understand your customer through digital channels will determine competitive advantage.

Engaging your customer through applications allows you to deeply understand each and every customer through his or her unique set of interactions — all of which can be recorded and analyzed. This is deeply transformative; the notion of a “generic” or “average” customer is a thing of the past.

You don't have to send out surveys to determine how well you are satisfying your customers; you can determine that directly through measurement and analysis. Are performance or responsiveness issues causing people to abandon your app? You can measure that. Is the new feature you just rolled out being discovered and adopted? You can measure that, too. Are you retaining customers with your app over time? Understanding customer churn is essential. Better yet, you can use advanced analytical techniques to predict customers who are likely to churn and to intervene beforehand.

It's important to create a complete picture of the customer experience. This means understanding the user experience from the application level all the way through to the back-end services and operations that power the app. For example, you may find that users are not able to find a particular feature due to lack of discoverability in the application's user interface. Or you may discover that users find and try the new feature but abandon it because of ease-of-use challenges. Or it may be that users stop using the feature because of performance problems, or crashes due to code quality. You may see that the feature is having performance problems at certain times of the day, or is being used less on certain devices. Ultimately, the question you're trying to answer is why the feature isn't meeting customer need, and finding the answer is essential to be able to make the right changes needed to successfully deliver the desired experience and value to the customer.

Creating a picture of your customer experience requires instrumentation all the way from the user interface to backend services. For example, you can instrument the app's UI to measure frequency and duration of use of each feature in the app. You can instrument service response times for APIs, network utilization, or database access patterns. There is almost nothing you can't instrument in the flow of the software delivered to the customer, and care must be taken to instrument the things that add truly *meaningful* data — data that can contribute to your understanding of the user experience without swamping you in noise. Choosing what to instrument often requires collaboration across multiple disciplines — including those outside technology. For example, the marketing team may want to know whether one arrangement of product offerings presented in the company's app is more effective than another in generating customer interest. The development team can implement a feature toggle in the software allowing two different populations of customers to experience the two approaches simultaneously. The results can be measured and analyzed to determine not only the more effective approach but also to generate additional insights such as demographics-specific differences revealed in the outcomes.

The digital experiences you deliver are not static. Agile methods and a modern software factory optimized for rapid delivery enable you to experiment, analyze, and constantly deliver new value to customers. To be fully effective, this process must be a cycle that feeds data-driven insights back into the system as an integral part of continuous improvement. Monitoring and analysis of application performance never stops. From an operational perspective, it must not only quickly catch unanticipated problems and identify areas negatively impacting customer experience, but it must also provide the needed insights to fix such problems as quickly as possible. Measurement and analysis also

creates insights into customer behavior and engagement as well as the ability to understand differences between expected and real-world outcomes.

ENABLING CONTINUOUS CUSTOMER FEEDBACK ON APP PERFORMANCE AND EXPERIENCE

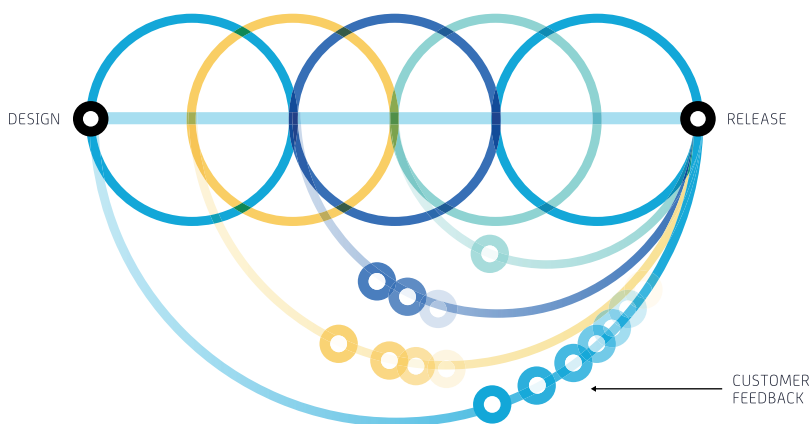


Figure 7-1: A modern software factory will enable you to measure and analyze your engagement with your customers. Data-driven insights are part of a cycle that fosters continuous improvement in the experiences you deliver. Your software can collect and analyze data not only to inform how your applications are performing and to guide continuous improvement of your experiences, but also to provide deep business insights into the relationship between your customers and your brand.



Applications make it possible to connect directly to your customers, giving you a powerful tool to continuously measure and analyze your effectiveness in meeting their expectations. Understanding how your applications are performing with your customers will give you valuable insights that will help reveal unmet needs and innovative ideas that can put you ahead of your competitors. This includes the ability to understand the customer journey through the software experience — the paths they take, the functionality they use most, and friction points they encounter — that can lead to insights resulting in new features, services, or improvement. Your digital connection creates an opportunity to understand the relationship between your company and your customers. You can then measure not only the responsiveness of your software experiences to your customers' needs, but also the responsiveness and adaptability of your company as a whole.

Common Challenges with Metrics and Measurement

Digital transformation requires rethinking how, why, and what is measured. The traditional IT focus on operational metrics and infrastructure efficiency won't scale to meet the demands of an experience-centric approach focused on rapid business evolution. Operations will, of course, continue to be a critical part of the picture, but the new focus must include external outcomes as well as internally focused operational and efficiency improvements. In addition to factors such as cost and complexity, improvements must also be measured against the impact on the customer experience. Perhaps some additional cost or increased complexity is justified to improve the customer experience in a meaningful way. And data-driven approaches broaden the scope of operational improvements that are possible. For example, using data to dramatically increase the effectiveness of fraud detection and prevention can also have a transformative impact on an entire business.

Just as with other areas, digital transformation requires metrics and analysis to be integrated into the entire software process from design to delivery. All the disciplines involved in the process of delivering digital experiences will have a stake in measurement and analysis as part of the critical feedback loop needed for continuous improvement.

Not surprisingly, this change in the scope and role of measuring, monitoring, and analyzing with a focus on customer-facing outcomes brings with it a whole new set of challenges:

- **New skills and tools.** Extracting meaningful insights from varied sources of data goes far beyond traditional reporting of operational metrics and statistics. The acquisition of new tools and new skills may be required to be able to move beyond dashboards and to generate business and customer experience insights.
- **Breaking down silos.** Realizing the full potential of measurement and analysis will require cross-functional collaboration within and around the technology organization. Instrumenting and analyzing customer adoption of a new feature will often require participation of design, development, test, data science, and operational disciplines. It will likely also include collaboration with the sales and marketing teams to ensure that the entire business can gain insights to inform actions.
- **Constantly changing environment.** As new apps and new features are developed and deployed, measurements and the questions they aim to answer will change over time. This will be a major departure from monitoring stable, slowly changing environments and will require navigating through cultural change.
- **Insights instead of data.** A common response elevating the importance of measurement is to measure too much, or to measure the wrong things. Be wary of the temptation to over-instrument code and infrastructure to the point where you are collecting massive amounts of data that only becomes another management headache. The reason for instrumentation, measurement, and analysis should always be grounded in insights that create meaningful value to the business.

It will be important to avoid the trap of metrics for their own sake. Generating impressive-looking graphics and charts can provide a false sense of security and the illusion of knowledge and control. Creating an agile analytics mindset focused on providing business insights and customer value will require iteration with concrete, actionable results as the goal.



Getting Started

Metrics, data, and analysis must answer “why,” not just “what” to ensure continuous improvement of customer experience. Why is a feature not being used? Why are some customers churning off an application? Why is a minor improvement driving a spike in adoption? As with agile, prioritizing customer value through measurement requires both organizational and cultural transformation. Here’s a blueprint for starting that process:

1. Focus on the questions.

The questions you are working to answer must become your true north. Instrumentation, data collection, and analytics are simply a means to an end. Technology is a powerful tool, but it can’t tell you the questions that matter to your customers and your business. Importantly, make sure that the answers to your questions drive specific, meaningful action.

2. Measure for insights.

It will be tempting to over-instrument code and infrastructure, and to collect massive amounts of data that simply becomes another management headache. Resist the urge and start small instead, concentrating on a specific area to demonstrate value. Keep in mind that you won’t always know what data will be valuable ahead of time and that you will need to be speculative. Don’t be afraid to over collect strategically as long as doing so does not become an operational burden.

3. Run experiments.

Use analytics to guide ongoing experiments and to predict changes that will improve the customer experience. Armed with insights, you can then make changes to your service — or provide two different versions at the same time — and measure customer response to inform your product development. Analytics allows you to validate the results of proposed optimizations before adopting them at scale. You must not experiment at your customers’ expense, but you can learn from your customers to improve the value and experience you give them.

4. Optimize for the business.

Ensure that your efforts are deeply integrated into the business. The insights gained from targeted instrumentation can inform your business, and your business needs should inform the insights you set out to obtain. Your digital channels are central to your ability to understand your customers who, in turn, help to shape the evolution of your products and business. Integrate customer-facing measures such as Net Promoter Score (NPS) to track your progress and to help guide the evolution of your applications and services.

Don't get distracted by data. Instrumentation and data collection are important but must be focused on answering questions that matter to the customer and to the business in a continuous feedback loop.



Partner Perspective

DHL helps clients transform their businesses by putting data to work in new ways²

Data is everywhere in the software-driven enterprise. The challenge is to harness its power without being overwhelmed by the complexity and scale of the job. For DHL's supply chain, helping customers harness data is a core business imperative. That might mean analyzing client data for useful feedback on process improvements, or distributing data down to levels of the organization that have not had access to it before.

In many cases, extracting value from data depends on organizing and analyzing it the right way to create new insights. In one instance, DHL reconfigured the supply chain of a consumer electronics company by segmenting the firm's sales information into smaller and smaller sections based on categories of devices. "By cutting and slicing the data, we were able to work out a much more efficient way of configuring their supply chain. Once we got the data very granular, we could pick up the pattern that some items have a certain profile of behavior. So, we were able to use that data more intelligently," said Damian Pike, DHL's Vice President of Innovation.

A division of the Bonn, Germany-based Deutsche Post DHL Group, which had \$64 billion in annual revenue last year, DHL Supply Chain gleans wisdom from many sources. Take, for example, data that comes through clients' social media feeds. DHL scrutinizes comments to understand how customers are experiencing the supply chain. It then turns that data into a source of continuous improvement for the client, which can refine processes through a series of test campaigns. "That is the sort of activity you would not have imagined happening five years ago in supply chain, but is very much the situation right now," Mr. Pike said.

8

Innovating to Deliver Customer Value

Today, software is a main driver of business innovation — the two are inextricably linked. Although it may be possible to have software innovation without business innovation, it's getting harder to imagine having true business innovation today without software innovation. Understanding what innovation is — and what it isn't — is key to focusing your resources on those activities that drive differentiation and growth.

Although the word innovate comes from a Latin word meaning to renew, many confuse innovation with invention. A simple rule-of-thumb is that invention is something that happens when you don't know what you want to happen, and innovation is what happens when you do. There's generally a purpose behind innovation, and the truth is that most of it is incremental — the majority of it, in fact. Digital innovation, then, is about evolving the capabilities, tools, and culture that enable you to renew your business with technology — primarily software — as an enabler of your goals. Unless you're a start-up with just a handful of people, or an advanced research lab, you're going to get a lot further looking at innovation in terms of “evolution” rather than “revolution.”

Since no business has unlimited resources, a basic challenge is fostering innovation that truly matters. History is replete with companies that were innovative but failed to differentiate themselves in the marketplace. Being an innovator means being sensitive to the balance between raw ideas and their pragmatic application to real customer problems and unserved or underserved demand. This chapter is all about finding that balance.

Finding the Sweet Spot

Focusing on innovation that matters to the business may challenge technology organizations and the businesses they support. Different businesses will have different innovation risk preferences, structures, and cultures to support it. Thomas Edison famously said: “I was always afraid of anything that worked the first time.” At the same time, when he opened his lab in New Jersey, the goal he gave himself and his employees was reportedly to produce a minor innovation every week and a major one every month. Edison understood the importance of speed in discovery *and* commercialization. He was a genius, but he was also aware of the practicalities of building a sustainable business.

Quantifying your risk-return trade-offs in as much detail as possible is central to ensuring that you are working on the projects that will deliver value to your customers. Every business in the application economy must consider how they can implement the latest technologies to innovate without losing focus on customer demand in the here and now. Striking that balance between current operations and future innovations is critical to business longevity.



As the pace of digital disruption accelerates, businesses in every sector are looking to shorten software development cycles, increase deployment frequency, and ensure an optimum customer experience. Why? Because in the digital-first world, getting to market quickly with an innovative experience for your customers can put distance between you and the competition. This is why every business today must also be a software business. Creating a competitive advantage with software requires software development to become a core competency, and mastering the intersection of innovation and speed to market will separate the winners from the also-rans.

Rebalancing investments to create new value

Despite clear need, most businesses can't afford to budget for digital transformation and innovation as a completely new, independently funded endeavor. Although some additional investment above and beyond existing technology spending may be required, optimizing existing spending should be the first step in creating new capacity to fund innovation-targeted transformation. In technology organizations where the majority of resources are dedicated to “keeping the lights on,” restructuring and reprioritizing investment across the entire portfolio can create significant savings that can be redirected to fund existing and new projects with the greatest business potential.

An integrated approach that evaluates the entire technology portfolio is an effective tool for funding new investments within an existing budget, and it also ensures that the impact of investment changes is thoroughly understood. For example, a new API initiative or application development effort may require additional investment in a specific legacy system rather than resource reduction. Or you may find that rebuilding some existing

functionality will yield significant savings that can in turn fund new initiatives. Investment in automation in particular can yield substantial long-term savings that can be applied to value-creating efforts.

Most importantly, you must be able to *objectively* evaluate each and every project based on the value that it provides in the current business context. Stakeholders may have asked for some technology capability many years ago; that effort may still be funded at its original level — or perhaps higher — even though the need may no longer exist or has decreased in priority. Technology projects have a tendency to take on a life of their own; “pet projects” can be a seemingly endless draw on resources. You may find that you have a collection of zombie projects where the original justification either no longer exists or is lost to history. These and other low-value efforts represent opportunity cost that detract from your ability to invest in innovation that really matters.

The need for ongoing, dynamic portfolio evaluation is essential not only to rebalance legacy investment but also to ensure that the entire investment pipeline — including new innovation — is achieving desired business outcomes. In fact, the need for continuous and rigorous assessment is especially critical at the front end of the innovation cycle where new ideas are not yet fully proven. Not all promising ideas will translate to proven business value. Changing market conditions, technical feasibility challenges, and lack of customer interest are just some of the reasons to reconsider an investment decision and to potentially make room for more promising projects.

INNOVATING TO DELIVER CUSTOMER VALUE

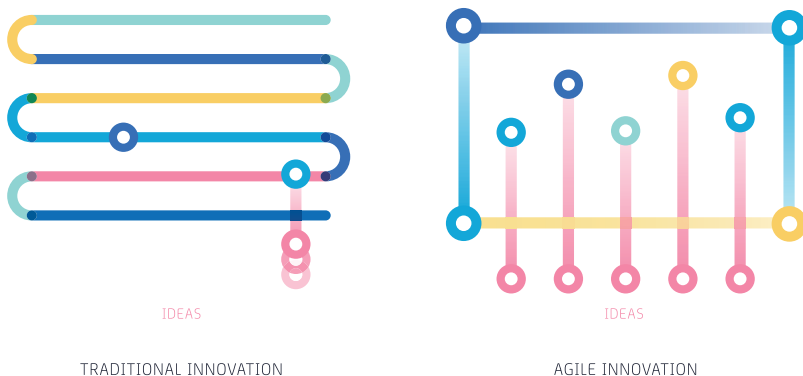


Figure 8-1: Removing barriers to innovation requires a deep understanding of how ideation in your organization works, how it can be amplified, and what processes are in place to nurture new ideas. Achieving a balance requires filling your innovation pipeline, and also quantifying your risk-return trade-offs to ensure you are working on projects that will deliver value to your customers.



Partner Perspective

HCL on how 21st Century enterprises manage technical debt

Kalyan Kumar (KK), Executive VP & CTO, HCL Technologies, shares his thoughts about how enterprises should deal with technical debt.

We are on the threshold of yet another technology revolution, the “augmented intelligence revolution.” At a time when machines are being upgraded with artificial intelligence and robotics, we are simultaneously extremely focused on providing the customer with a consistent end-to-end experience. We’re experiencing not just a technology revolution; we’re also experiencing the transformation of the way customers consume.

The service-oriented customer is accustomed to constant upgrades of consumer experience in his personal space and expects the same in the enterprise space. This in turn pushes the IT organization to innovate constantly. However, managing technical debt across existing legacy architectures and systems requires extraordinary effort and, as a result, organizations are becoming risk averse with innovation.

To be a 21st century enterprise, you need to think like a start-up, while acting with the maturity of an enterprise. Strategy needs to be charted out carefully; it ought to be based on a consultative approach so that change can be managed across a well-defined lifecycle. To mitigate technical debt and encourage a culture of innovation, HCL recommends the following approach:

- Identify and quantify technical debt
- Determine whether the benefit of overhauling existing architectures/systems exceeds cost over the expected life of the system
- Define the process to leverage multimodal IT
- Define an organizational change management plan to encourage the agile and DevOps culture in the organization
- Identify the current state of automation and plan the extreme automation journey
- Allocate the innovation budget after taking the above into account

Applying a 21st century automation strategy to an IT environment where standardization is yet to be achieved can create challenges. Therefore, it’s critical to get the foundation right. Much of innovation around these areas is becoming the hygiene of 21st century enterprises, and the lack of strategic planning for such initiatives is the invitation to failure. First, analyze it. Plan it. And then, go for it.

Culture and organization are core to the rebalancing process

Achieving effective, ongoing reviews of your technology portfolio is likely to require culture change. This is especially true in organizations that have been operating in a stable, steady state for a number of years. People will need to become comfortable with an openly inquisitive and fully transparent approach that questions assumptions that previously may have been taken for granted. Does a project still make sense? Should we decrease investment in one project to be able to increase investment in another project that has higher business value? Is the portfolio of our projects optimal for our business? Managing through project size reductions and exits requires thoughtful management. Such rebalancing must not be seen as punishment, and care must be taken to ensure that talent can be redeployed to other projects.

Customer engagement is also critical to achieving an optimal balance. A shorter, tighter feedback loop with customers allows customer feedback to dynamically inform both product development and investment decisions. Perhaps you're building the wrong thing and need to stop or pivot? Your organization needs to become comfortable seeking data and using it to adjust priorities, and working dynamically instead of rigidly following a plan that leads to a low payoff. In other words, a positive outcome trumps perfect execution of a bad plan.

Ongoing portfolio review should also include architecture as a core pillar to ensure that technology development and planning is coordinated across the entire portfolio. Are there common building blocks that could be combined to reduce complexity and maintenance overhead? Are the right platform-level components being built to connect applications and platforms? Is the right technical debt (see the sidebar earlier in the chapter) being paid down? Addressing technical debt by targeting high-value areas is critical to both freeing up resources and enabling rapid innovation. Such architectural coordination will require working effectively across the entire technology organization and breaking down silos.

Both your culture and your organization need to be optimized for rapid, continuous innovation. There are many approaches to building a technology innovation pipeline. Some argue for the benefits of creating separate "skunkworks" projects that operate outside of existing structures and processes, or building an independent organization working in parallel to help quarantine innovation from the pressures of daily business. Others advocate for directly embedding innovation across the entire technology organization. The messy reality is that you likely need to mix and match approaches based on your specific situation. Although creating separate groups or pockets of unique expertise may be needed to jumpstart innovation and software development in new areas, your long-term goal should be to evolve the entire technology organization to operate at a new level of effectiveness with a shared responsibility for innovation in every aspect of technology. Be wary of creating a permanent innovation group separate from the rest of the development organization; digital transformation needs to include everyone to be fully effective at scale. Why? Because separate, insulated efforts that focus exclusively on new innovation miss the

larger opportunity to incorporate digital transformation across the entire technology portfolio. You must optimize locally — within product teams and business units — and globally across the entire development organization.



Getting Started

Making the changes necessary to achieve rapid and sustained customer-focused innovation will require patience and determination. You will need to keep in mind that change is disruptive; make sure that the outcomes you aim to achieve are clear and measurable as you proceed. To make sure that happens, be sure to do the following:

- **Conduct an innovation audit.** Understanding your current state is key to understanding the distance between you and your ideal innovation state. Every organization has a different risk tolerance for taking on new projects. For example, is the bulk of your innovation incremental? What's the return you expect on incremental innovation in the marketplace? Where is the competition? Where do you want to push the envelope? These and a host of other questions around your innovation engine such as available skills and talent, funding, and how innovation is connected to your company's strategy need to be considered. An internal cross-functional team can start this assessment to identify innovation gaps and opportunities. If you're not sure that you'll get an honest picture of where you really are, there are external resources expert at these assessments who can get you started.
- **Perform regular portfolio reviews and dynamically adjust investments.** Digital transformation is driving rapid changes to business; you must dynamically adjust your technology investments to reflect changing business priorities. This is not just a question of "legacy" vs. "new"; you may find that you can unlock new efficiencies or value by increasing an investment in an existing area, and you may need to decrease or perhaps halt investment in *new* areas that are not delivering value. You must be willing to question assumptions and follow the data. Engage directly with your business stakeholders in this process so that you and your team have a full understanding of the business context of everything in your portfolio. And don't limit portfolio adjustments to annual budgeting. At a minimum, conduct quarterly reviews to be able to shift investments as conditions dictate.
- **Drive cultural change to support rapid innovation.** It's important to understand that it's virtually impossible to disconnect what you do in an organization from how you do it. Innovation is not likely to come from doing things the way they've always been done in the past. Leadership's cry that "We need to be more innovative" is really a statement about a competitive landscape where others are outpacing you with innovative products or services. The challenging part is translating that cry into the organizational and culture change required to drive real innovation at the center of your business. Top-down culture change programs driven through executive mandate can be rolled out quickly at scale but miss the mark when they fail to address the underlying culture that is the sum of behaviors of the organization.

The best way to introduce innovation is to start small. Take a visible project and change the rules around behaviors — such as creating a safe environment or “sandbox” where innovators can share their ideas without fear of being reprimanded for failure. Measure the results, and grow the model organically across the organization by building on success and leading by example. Also, when you implement programs like agile, they must come with ready-made boosters built in to protect, incentivize, and support innovators. Agile concepts such as value-orientation, rapid iteration, and cross-functional teams can give you an innovation lift.

- **Use tools from the start-up world to guide you.** Innovation is like a muscle; it needs both structure and practice to become strong, as well as a supportive environment in which to thrive. Start small — build a simple innovation catalyst such as an internal incubator, and gather and support a handful of good ideas. Adopt lean start-up approaches. Have a small advisory team help intrapreneurs assess the pace of learning and validation of assumptions using a “persevere, pivot, or pause” assessment on a monthly cadence. Use a lightweight lean business model canvas to evaluate and manage your innovation portfolio. Focus on Pirate metrics (Acquisition, Activation, Retention, Referral, and Revenue, or “AARRR”) that assess customer and market demand. And most importantly, be disciplined about terminating pet projects that consume precious resources; the savings can be applied to high-value efforts that may otherwise be starved for funding.



Partner Perspective

Orange uses software innovation to advance its business¹

It was nearly a decade ago that the company then known as France Telecom — now Orange — began its journey to becoming a software-driven enterprise. “We were just trying to understand what the benefits could be of bringing some apps into our information technology group,” said Patrice Slupowski, Orange’s Vice President for Digital Innovation. “At that point, these applications were very much considered a foreign element inside our company.”

Today, Orange thrives in the app economy. It has a proprietary system to monitor the quality and performance of its internal development process and a facility called The Techno Center, where its biggest software development projects are carried out in an agile mode. Of course, the company still has vital legacy systems. But given its size — €39 billion in annual revenues (2014), 155,000 employees and 247 million customers worldwide — Orange is moving at what feels like breakneck speed.

That is perhaps the biggest transformation at Orange since its immersion in the app economy — the velocity with which the organization generates and implements ideas. “We are quite obsessed now with the feeling that we have to go fast; that we have to either deliver fast or fail fast on any project. That way, we are able to quickly learn from failure,” said Mr. Slupowski.

(continued)

(continued)

His team is especially focused on open-source innovation, gathering feedback from French developers on how applications are used. This allows the company to have a significant impact on research as well as the development of patents, which also helps sharpen its competitive edge.

But becoming a software-driven enterprise took time, and progress came in waves. Orange technologies would see an app being used by another company and create a prototype to try in its own projects. One of the biggest issues — even when a prototype worked well — was deciding whether to implement it on a companywide basis. Orange was inspired by the first APIs developed by the Internet giants and tried to adapt some of their work for its own services.

“But then it would conflict with other things we were doing internally, and people were afraid we were cannibalizing our own activities,” Mr. Slupowski said.

Still, the evolution continued. Explanations and analysis of the advantages of advanced software methodologies and application development accumulated. Eventually, the CEO got on board producing a predictably galvanizing effect. “When the head of the company explains why it is important to move in a certain direction, that simplifies everything,” said Mr. Slupowski.

Now, Orange is borrowing ideas and approaches — from lean start-ups in particular — and assimilating them into the organization to increase its agility. “Globally, we consider that innovation is a key differentiating factor,” Mr. Slupowski said. And the company backs that up by having several thousand people dedicated to the task of innovation — a tangible sign of Orange’s progress in the years since the company launched its first set of APIs.

Conclusion

In a world of pervasive disruption that demands continuous business model renewal and innovation, getting into the software business and undergoing some level of digital transformation are now table stakes for staying competitive. There are many roads to becoming a software-powered business and the path you choose will depend on a number of factors, such as your current and anticipated competitive landscape, your existing technology footprint, your desired level of customer intimacy, your growth objectives, and the resources at your disposal for investing in software capabilities.

Whichever path you ultimately choose, your journey will include key common elements. First, you must turn to automation and highly integrated development to ensure a steady flow of software and new value to your customers. Second, you need to acquire agile skills not native to most organizations accustomed to old-style organizational structures and heavyweight processes. Achieving true agility requires you to deeply re-architect how you operate with a focus on team-based collaboration, ongoing learning, and customer outcomes. Third, digital engagement gives you new tools to understand your customers. Instrumentation and analysis not only give you the ability to continuously improve the experiences you deliver, but also allow you to optimize your business based on data-driven insights. Your customers will be the true North on your journey as you seek to engage them, understand them, and delight them.

The revolution taking place today may be fueled by software, but the real revolution is about embracing new approaches and methodologies that allow software to create critical new business value. As we said in the introduction, while the scope of the changes required for transformation may seem daunting, they represent a rare and compelling opportunity to re-think, reshape, and reinvigorate your business: to remaster it for a new digital future.

Afterword

By R “Ray” Wang

Founder & Chairman at Constellation Research, Inc.

We are in the midst of a digital business revolution. Organizations can sense that their very survival is at stake. They see the seismic shifts in play as industry after industry falls to upstart, agile disruptors. The business impact is measurably significant: to take one example, 52 percent of companies in the Fortune 500 have gone through mergers, acquisitions or bankruptcies since 2000. Even more disruptive, 55% of the Fortune 500 failed to make a profit in 2015. Meanwhile, the average age of an S&P 500 company is down from 60 years in 1958 to less than 15 years today.¹ That is a 400 percent compression in the average age of a company, creating fundamental and permanent change across the global business landscape.

Today, digital winners embrace software to drive business model differentiation. While the adoption of packaged software has improved operational efficiency in organizations, the commoditization of best practices resulting from widespread use of technology has reduced or even eliminated the competitive advantage to be gained. With the rise of innovative new digital businesses — from traditional or completely new competitors — organizations must invest in their own digital capabilities to build differentiation. Incremental innovation is not enough. Transformational innovation is required just to stay in the game.

So, make no mistake; the end result of this shift is not a digital divide. Organizations face a ‘winner take all’ market. In almost every industry segment, the top three players have taken 40 to 70 percent of the market share and 40 to 77 percent of the profits.² The winners have one thing in common: they adopt a digital-first approach to their strategy. Digital transformation is no longer an option — it’s a necessity. Simply adding a new technology or adopting the latest trendy methodology is not enough. Early adopters realize the transformation must happen at the business model level. Leaders must deliver a new way of thinking about their core businesses — one that starts with driving digital into their business DNA. Only this will enable them to deliver the experiences and outcomes their customers are looking for, and to move from selling products to keeping brand promises.

In fact, organizations not natively born digital are poised to disrupt digital natives. Why? Market leaders have often had to build the methodologies, tools, and technologies from scratch to gain competitive advantage. The ability to easily plan, build, manage, secure, and iterate software is a foundational requirement in digital transformation. As these methodologies, tools, and technologies improve with time, fast followers focus on scale in order to close the gap with market leaders. Fast followers can overtake market leaders as long as they commit and invest in strategic software development capabilities and agile development methodologies.

While all this talk about digital can sound technical, let's be clear that the point is to improve the human experience and cultivate digital artisans. Digital by nature can seem hard and cold, and yet it's true that we need the best cold, hard math to advance more sophisticated algorithms. As everyone knows, the ultimate goal of managing large data sets is to improve decision-making. Since data is the foundation of these digital businesses, having better algorithms will drive our ability to automate, predict, and even augment decision-making.

At the same time, by fostering design that drives the modern human experience, we need to develop digital as an art-form. Having the best math may lead to great spreadsheets — and tons of numbers without context. Pairing the best math with the best design means we take into account the human component. So where we often hire for science, technology, engineering, and math, we also need to account for anthropology, ethnography, design thinking, user experience and storytelling. The goal? Bringing the right-brain and left-brain capabilities together and building a world that, instead of denying our humanity, actually embraces it with the rise of digital artisans.

There are other important considerations in this new world. For example, the shift from direct ownership of computing power, data, and insights to ownership that is networked accessed opens up organizations to cyber-threats. Subsequently, organizations must proactively address security. Meanwhile, organizations can expect regulatory requirements from governments will only increase. To build digital into your business DNA, you must incorporate a continuous and comprehensive compliance system with digital design and security considerations built in from the start.

Organizations that are digitally remastered will emerge as the winners in the digital era. Transforming business models, keeping brand promises, and humanizing digital technology with the work of a new breed of digital artisan provide the foundation for success. However, those who invest in transformational innovation must translate their business models into software. Without successfully differentiated software, there can be no digital transformation. To update Marc Andreessen's famous phrase, software not only eats the world, but it increasingly IS the world.

ENDNOTES

Chapter 2

1. Williams-Sonoma Rethinks "IT", Williams-Sonoma interview conducted by CA Technologies, November 2015

Chapter 3

1. Lockheed Martin's uses DevOps to transform software development, CA Technologies & Oxford Economics, 2015, [http://rewrite.ca.com/content/dam/rewrite/files/White-Papers/CA App Economy - Lockheed Martin case study - August 10 2015 final.pdf](http://rewrite.ca.com/content/dam/rewrite/files/White-Papers/CA_App_Economy_-_Lockheed_Martin_case_study_-_August_10_2015_final.pdf)

Chapter 4

1. Business Agility Survival Guide, <https://www.rallydev.com/resource/business-agility-survival-guide>
2. The High Cost of Low Performance, http://www.pmi.org/~media/PDF/Business-Solutions/PMI_Pulse_2014.ashx
3. David Frico, "What is the Role of agile vs. Traditional Methods? An analysis of XP, TDD, Pair Programming, and Scrum (Using Real Options)" (<http://www.davidfrico.com/rico08b.pdf>)
4. A CIO's Guide: Five Steps to Getting Started with Business Agility. (<http://www.ca.com/content/dam/ca/us/files/white-paper/a-cios-guide-five-steps-to-business-agility.pdf>)
5. Nies, Zach et al. (2014) The Enterprise Lean Startup Approach, Rally, 2014 <https://www.rallydev.com/toolkits/enterprise-lean-startup>
6. PayPal takes a big-bang approach to agile to increase speed to market, PayPal interview conducted by CA Technologies, November 2015

Chapter 5

1. Source: ProgrammableWeb, as of 2/23/16 (<http://www.programmableweb.com/>)

2. "The Strategic Value of APIs", *Harvard Business Review*, 1/7/15 (<https://hbr.org/2015/01/the-strategic-value-of-apis>)
3. "APIs and the Digital Enterprise", Freeform Dynamics and CA Technologies, November 2015 (<http://rewrite.ca.com/us/articles/application-economy/apis-and-the-digital-enterprise.html>)
4. Powering L'Oreal's digital transformation through APIs, "APIs: The Building Blocks of the App Economy" Infographic, CA Technologies, 2016, <http://paidpost.nytimes.com/ca-technologies/apis-the-building-blocks-of-the-app-economy.html>.

Chapter 6

1. "Everything You Need to Know About the Internet of Things", Jacob Morgan, Forbes.com, Oct 2014, www.forbes.com/sites/jacobmorgan/2014/10/30/everything-you-need-to-know-about-the-internet-of-things/ - 46d9cc6a5af3.
2. "Software — The New Battleground for Brand Loyalty", CA Technologies, 2015, <http://rewrite.ca.com/content/dam/rewrite/files/White-Papers/software-the-new-battleground-for-brand-loyalty-final.pdf>.
3. "Home Depot: Will the Impact of the Data Breach be Significant?", Forbes.com, Trefis Team, Mar., 2015
4. "\$10 Million Settlement in Target Data Breach Gets Preliminary Approval", NYTimes.com, Hiroko Tabuchi, Mar. 2015, www.forbes.com/sites/greatspeculations/2015/03/30/home-depot-will-the-impact-of-the-data-breach-be-significant/-443d96f969ab
5. "What Data Breaches Now Cost And Why", "Dark Reading, May 2015. ([http:// www.darkreading.com/attacks-breaches/what-databreaches-now-cost-and-why/d/d-id/1320578](http://www.darkreading.com/attacks-breaches/what-databreaches-now-cost-and-why/d/d-id/1320578))
6. "The Advisory Board Company Helps Healthcare Professionals Securely Unlock Data Through Effective API Management", CA Technologies and Zogby Analytics, 2015, www.ca.com/us/collateral/case-studies/the-advisory-board-company-helps-healthcare-professionals-securely-unlock-data.html

Chapter 7

1. “The Internet of Things: Mapping the Value Beyond the Hype”, McKinsey Global Institute, 2015, www.mckinsey.com/business-functions/business-technology/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world.
2. Case Study: “DHL helps clients transform their business by putting data to work in new ways”, CA Technologies & Oxford Economics, 2015, http://rewrite.ca.com/content/dam/rewrite/files/White-Papers/DHL_Case_Study_final.pdf.

Chapter 8

1. Case Study: “Orange uses software innovation to advance its business”, CA Technologies & Oxford Economics, 2015, http://rewrite.ca.com/content/dam/rewrite/files/White-Papers/CA_App_Economy-Orange_case_study-July_14_2015.pdf

Afterword

1. *Disrupting Digital Business*, R “Ray” Wang, Harvard Business Publishing, May 2015
2. *Disrupting Digital Business*, R “Ray” Wang, Harvard Business Publishing, May 2015

Back Cover

Gartner, 2016 CIO Agenda: Global Perspectives on Building the Digital Platform, 25 February 2016

[illegible]

[illegible]

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

AFTERWORD BY R "RAY" WANG, INDUSTRY-LEADING ANALYST
AND AUTHOR OF *DISRUPTING DIGITAL BUSINESS*

DIGITALLY REMASTER YOUR BUSINESS

In the new app economy, the real disruption is not the technology but your customers' expectation for continuous, high-fidelity digital interaction. Discover how to integrate software into the very core of how your business operates and engages with customers.

"LEVERAGING THE POWER OF SOFTWARE, MARKET LEADERS AND CHALLENGERS CAN EXPLOIT BROADER TECHNOLOGY AND CUSTOMER TRENDS TO BRING WHOLE NEW CLASSES OF DIGITALLY-ENABLED OFFERINGS TO MARKET."

Dale Vile, Research Director, Freeform Dynamics

"WE ARE NOW DEEP IN THE ERA OF DIGITIZATION. ALL CIOs ARE EXPECTING THE DIGITAL INTENSITY OF THEIR ENTERPRISE TO INCREASE OVER THE NEXT FIVE YEARS, WITH PRIVATE-SECTOR CIOs PREDICTING DIGITAL REVENUE TO GROW FROM 16% TO 37% IN THE NEXT FIVE YEARS, AND PUBLIC-SECTOR CIOs PREDICTING A RISE FROM 42% TO 77% IN DIGITAL PROCESSES OVER THE SAME PERIOD."

Gartner, February 2016, Report #G00297534



As chief technology officer of CA Technologies, Otto Berkes is responsible for technical leadership and innovation, further developing the company's technical community, and aligning its software strategy, architecture and partner relationships to deliver customer value. A 25-year industry veteran and an early champion of mobile computing, he has extensive experience leading the development of cutting-edge products and mobile technologies.



ISBN: 978-1-119-27153-6

ALL PROCEEDS
TO BENEFIT

CODE *for*
AMERICA

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.