Praise for
UNBUNDLING THE ENTERPRISE

"To find novel value (digital treasure), CIOs must dive deep, beyond the surface of websites, and connect the dots in this ever-evolving network of possibilities. *Unbundling the Enterprise* will help you navigate these vast digital seas."

—**Esat Sezer,** cofounder of Ritmus, former CIO of Coca-Cola

"I recommend this book for business folks looking for the 'aha' moment on APIs, and who want to make the business case for APIs in their organization."

—**Mark O'Neill,** Gartner Chief of Research for Software Engineering

"A phenomenal read! Stephen and Matt captivate you from the start. This book will challenge you to re-evaluate your perspective on how you can differentiate yourself in the modern data ecosystem."

—**Jason Beyer,** VP Data & Analytics, Bridgestone

"It's no accident that Stephen and Matt have written a book that clearly lays out a path for success in the digital economy. For years, they have been helping organizations drive new business value through the use of APIs. Not only do they show why the world's most innovative companies have a knack for creating repeatable digital success, but they also identify patterns that every business can emulate to reach their goals."

—**Steve Lucas,** Chairman & CEO of Boomi, author of *Engage to Win*

"In an age where the pace of change is faster than ever, this book offers a road map for leaders to harness the power of digital transformation, bridging the gap between high-level strategy and day-to-day operations. It's about creating a resilient organization that thrives on innovation, ready for the future's challenges and opportunities."

—**John Rowell,** cofounder and CEO, Revenium

"*Unbundling the Enterprise* provides a blueprint for leveraging APIs to not only achieve seamless integration and enhanced flexibility but also to foster an ecosystem of innovation that can significantly amplify your market reach and financial outcomes."

—John D'Emic, cofounder and CTO, Revenium

"*Unbundling the Enterprise* speaks to the imperative of digital data strategy for exponential value creation. Written in an approachable and easy-to-understand detail, it is an essential guideline for non-technical business leaders and executives responsible for growth."

—J. Ignacio Puente, CEO, Ulity, and VP Digital Transformation, Mobility & Subscription Platforms, Global Business Strategy & Operations, Santander

"Fishman and McLarty make an undeniable argument for making more intentional choices today that open a whole world of possibilities tomorrow. The value dynamics chapter alone is worth the price of admission, but I think readers across business, design, and technology will see the wisdom in these pages and find new ways to collaborate, strategize, and win together."

—Patrick Quattlebaum, CEO, Harmonic Design and coauthor of *Orchestrating Experiences*

"With powerful, real-world examples drawn from both digital disruptors and well-established incumbent firms in a variety of industries, *Unbundling the Enterprise* is an essential read for leaders striving to build an antifragile, thriving business poised for sustainable growth."

—Mahesh Motiramani, Head of Enterprise Customer Success, Workato

"I reckon this book will be on the desk of all digital leaders and aspirants soon. The new math weaves optionality, opportunity, and optimization beautifully. While value dynamics is a powerful technique, the economic framing of APIs through the lens of optionality is refreshing."

—Dinkar Gupta, Chief Information and Technology Officer, KPMG Switzerland

Stephen Fishman Matt McLarty

UNBUNDLING THE ENTERPRISE

APIs, Optionality & the Science of Happy Accidents

IT Revolution
Independent Publisher Since 2013
Portland, Oregon

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

First Edition
Printed in the United States of America
28 27 26 25 24 1 2 3 4 5 6 7 8 9

Cover design by Julianna Johnson and Kate Giambrone, Bologna Sandwich LLC.
Book design and illustrations by Devon Smith Creative LLC.

Library of Congress Cataloging-in-Publication Data
Names: Fishman, Stephen, author. | McLarty, Matt, author.
Title: Unbundling the enterprise : APIs, optionality, and the science of happy accidents / Stephen Fishman & Matt McLarty.
Description: Portland, OR : IT Revolution, [2024] | Includes bibliographical references.
Identifiers: LCCN 2024011612 (print) | LCCN 2024011613 (ebook) | ISBN 9781950508877 (paperback) | ISBN 9781950508884 (ebook) | ISBN 9781950508891 (pdf) | ISBN 9781950508907
Subjects: LCSH: Business--Technological innovations. | Application program interfaces (Computer software) | BISAC: COMPUTERS / Internet / Web Services & APIs | BUSINESS & ECONOMICS / Leadership
Classification: LCC HF5548.2 .F475 2024 (print) | LCC HF5548.2 (ebook) | DDC 658/.0553--dc23/eng/20240514
LC record available at https://lccn.loc.gov/2024011612
LC ebook record available at https://lccn.loc.gov/2024011613

ISBN: 9781950508877
eBook ISBN: 9781950508884
Web PDF ISBN: 9781950508891
Audiobook ISBN: 9781950508907

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

Dedication

Stephen—To my amazing wife, Kathleen Kelly, who inspired me to find my voice and made it possible for me to be of service to the world around me. To my parents, who helped me get a world-class education. To my children, Marina and Abigail, for making me a better person. And to my many friends and colleagues who helped me develop into a professional contributor and leader with something to offer the community.

Matt—For Chris, for her patience, for her support, for her love. For Daniel and Josiah, for reminding me what's important in life.

Contents

List of Figures

CHAPTER 7

CHAPTER 8

CHAPTER 9

CHAPTER 10

CONCLUSION

Prologue

Astounding Pirate Innovation

The Dread Pirate Greybeard stands at the helm of his famous ship, *The Legacy*. He steadies its course with his right hand. In his left, he clutches a rolled-up piece of paper: a treasure map. It took three years to track down the map, another year to hire the temperamental but uniquely skilled crew, and then the better part of another year to pack the ship and make it seaworthy. For the last month, they have sailed the treacherous seas. But finally—finally!—they see it. "Land ho!" yells a deckhand from the crow's nest. After executing their meticulous plan for five long years, they have made it to Treasure Island.

Two crew members drop anchor while the rest load rowboats with supplies: picks, shovels, and ropes. In the blink of an eye, the motley crew of pirates thrash their oars with excitement as a procession of rowboats makes its way to shore. The group forms two lines on the beach as the final boat arrives, carrying their fearless leader triumphantly to dry land. "Treasure!" he shouts, holding the map in the air. "Like I promised ye five years ago, there's no other option: with this, we'll all be rich!" The crew explodes in shouts of excitement and celebration.

Greybeard consults the map and then marches impatiently toward a distant rock. "The first marker," he mutters, his pirate crew in tow. From there, they venture through jungles, over quicksand, and under the shadow of a volcano until they finally reach a clearing with a shimmering lagoon. "We're close," Greybeard sputters. "I can almost taste it." Consulting the map, he

makes his way to a lone palm tree. Slowly and carefully, he takes ten measured paces to his left. Turning to his right, he takes twenty paces forward, bringing him to the edge of the lagoon. With a final turn to his left, he crawls the last twelve paces, his nose almost touching the sand. He stops suddenly, eyes bulging from his weathered face. The crew rushes to his side, desperate to see what has frozen their fearless leader. There, only slightly covered in sand, is an iron cross. Greybeard rotates it slightly so everyone can see it. *X* marks the spot!

"Dig, me hearties! DIG!" shouts Greybeard. Picks and shovels rain down on the area. Sand flies in all directions. Soon, the crew reaches clay, but nothing can slow their furious excavation. Within minutes, they are three feet beneath the ground. After a quarter of an hour, they've almost dug a lagoon of their own. "Keep digging!" says Greybeard with a slight quiver. "It's here. I know it. The map got us this far. It can't be wrong!" But hours more of digging doesn't turn up any treasure. Only the bones of those who had dug before. Demoralized, Greybeard and his crew retreat to their ship at sundown.

In the morning, Greybeard and his crew awaken to an unusual sight. Beached on the island is a pirate ship the likes of which they have never seen. The Jolly Roger flies from its main mast, but this vessel appears to be a collection of rafts roped together, wider than ten pirate ships combined, and completely covered in wooden crates. The puzzled pirates look on as a masked, black-clad figure emerges from the center of the super raft. "Good morning!" shouts the stranger. "The name's Bob. Fine morning for a treasure hunt, wouldn't you say?"

With that, the stranger tears a side off the first crate, letting its jangling contents spill onto the beach. From *The Legacy*, it's impossible to tell what the contraptions are that fall from the crate, but the crew sees Captain Bob pick one up and start winding it with a metal key. He puts the device down, and it scurries across the beach, stopping a hundred or so yards away. It starts digging furiously, creating a miniature sandstorm. Bob winds another device, then another, until all the devices have scurried to seemingly random spots on the island, digging relentlessly once they arrive. He repeats the process with every crate until it appears the whole island is covered with these mechanical pirates. Captain Bob walks casually toward his innumerable, unliving crew, surveying their handiwork.

DING! A loud bell rings out from behind a distant dune. Bob rushes toward the sound. "Aha!" he screams. "Long John's silver! I thought it might be here." DONG! A bell rings farther inland. "Whoa! I've never seen so many pearls in my life!" GONG! Farther still. "By Davy Jones's locker, it's the lost gold of Montezuma!" On board *The Legacy*, the crew stands silently, jaws agape as they watch a parade of treasure return to the raft, carried by a mischief of clockwork mice. Greybeard tears his map to shreds, then slowly throws its pieces one by one into the sea…

Introduction

There's treasure out there in the digital seas! But figuring out how to find it requires a deep understanding of the digital business landscape—both its opportunities and its obstacles. Since the World Wide Web took hold three decades ago, enterprises established before the dawn of the web have been trying to figure out how to navigate the new landscape and how they can map it to find digital treasure. These *digital settlers* have tried to apply the best practices of pre-web business, such as five-year plans and linear production lines, to their digital delivery but have yielded limited returns. Moving a large organization away from its status quo is hard, but establishing an effective new operating model is impossible if you don't understand the paradigm in which you are working. This is why so many established organizations have struggled with digital transformation.

Within that same thirty-year period, a new generation of web-native companies was born. These startups had no existing practices to apply to digital business. They were new companies aware that they were playing a new game. As opposed to viewing the web as a new channel to be added to their operating model, these *digital pirates* embraced the properties of the web as the essence of their business models: data as the raw material for creating digital products; high-speed, low-cost communication; and mass observability. Whether by necessity or design, these web startups were defining a new methodology for digital treasure hunting.

People first understood the web to be a collection of websites. Most companies—the digital settlers—moved quickly to create home pages that their customers could visit. But the digital pirates saw the web differently. Looking beneath the surface of the browser, they saw the internet as a way to easily and cheaply connect businesses to each other. Visionaries like Tim O'Reilly imagined the web as a network of digital services that any company could access. Early believers in this vision, like eBay and Amazon, made it a reality by creating APIs (application programming interfaces) for their digital products and services.

In the simplest terms, an API is an interface that allows software to talk to other software over the web. Unlike a GUI (graphical user interface), which is optimized to be used by humans, an API is optimized to be used by software programs. Developers create APIs/interfaces to deliver software functionality and data to other software programs (software-based consumers). Other organizations can then build software that accesses the API via network communication. As more companies published APIs on the web, and even more companies started using them, the vision of the web as a digital business ecosystem became real.

To understand how APIs power web experiences, consider a rideshare scenario involving two people: a rider and a driver. The rider orders a ride to the movies on their CloudCo smart speaker, checks the driver location and pickup time on their RideCo app, and then pays for the ride through the app when they reach the theater. The driver accepts the ride request through the RideCo app, locates the rider, and then follows the app's navigation to the destination. The list below illustrates what happens to enable these two user experiences, with APIs playing a big role (see also Figure 0.1).

1. The smart speaker sends the ride request message to CloudCo's backend services using an API.
2. The CloudCo backend then sends the message to RideCo's API.
3. The RideCo driver receives the ride request on their mobile phone via another API.
4. The rider and driver location services are enabled on the RideCo app by connecting to MapCo's geolocation API.

5. RideCo allows the driver and rider to contact one another through the mobile app, which is enabled by TelCo's communications API.
6. Finally, payment services are enabled on the RideCo app with the help of PayCo's payments API.

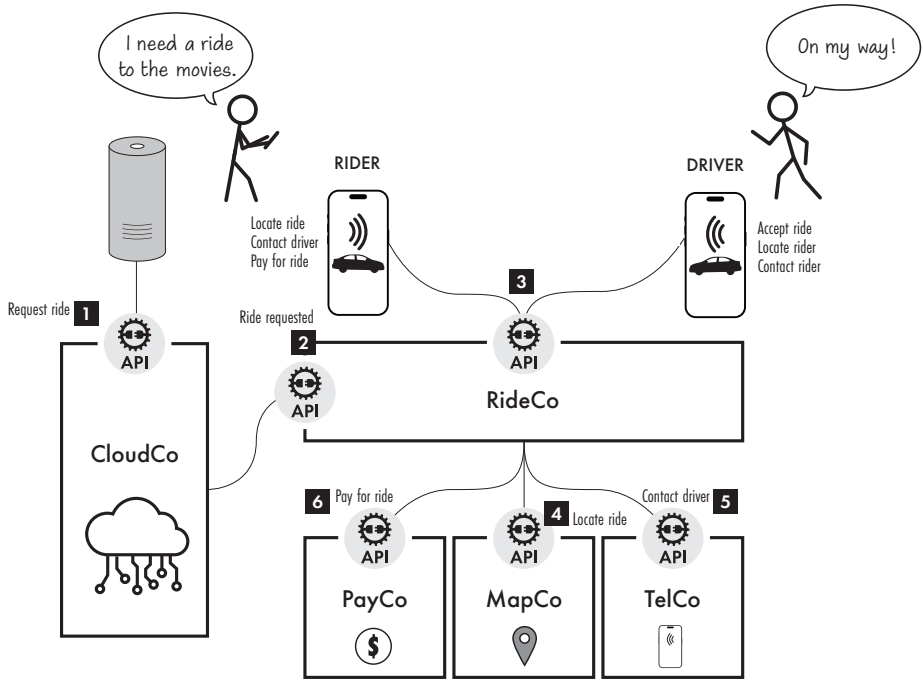


FIGURE 0.1: The Role of APIs in a Rideshare Scenario

The rider has a smooth customer experience, blissfully unaware that they just conducted business with five different companies (six if you count the driver as an independent contractor).

Not only does this example show the role APIs play in powering current digital experiences, but it also shows how a collection of API-enabled services allowed companies like Uber and Lyft to launch and scale in record time. Before the web and APIs, a company wanting to offer transportation services would need to create or buy its own payment software, mapping data, and

telecommunications services. With commercially available APIs like Stripe, Google Maps, and Twilio, those capabilities are instantly available. On top of that, both rideshare startups are able to run their applications through API-enabled cloud services, another variable that fueled the rideshare companies' fast start and rapid growth.

This rideshare example also illustrates the most important aspect of API-enabled digital treasure hunting: you don't need a map. Google didn't create Maps with Uber in mind. The same goes for Stripe and Twilio. These services were created as APIs, allowing them to be used in many unanticipated ways. Digital business is all about innovation, which means you can't draw up a detailed map to the treasures of tomorrow. We have no way of knowing what those treasures will be.

What digital pirates have shown over the last thirty years is that the right way to find digital treasure is to create as many options as possible in your digital landscape and then be ready to exploit new opportunities as they arise—prepare early, decide late. APIs help create options, and a high degree of optionality leads to unanticipated opportunities. To happy accidents.

There are many more examples of API options driving web innovation. The rise of social networks was API-enabled. A critical factor in Facebook's early dominance was its "Like" button, which was built on top of the Facebook Graph API.[1] Mobile apps were built on data and functions coming from cloud services via APIs. Even today's AI (artificial intelligence) boom is largely dependent on APIs. Generative AI is being incorporated into customer experiences through APIs, and corporations are embedding their own services into third-party models via API. All this API-enabled innovation begs the question: Are these digital pirates using some grand design or simply following their intuition?

Whether intentional or not, the good news is that there is a recipe for winning with APIs in the digital economy. This is the book that defines it and makes it accessible for legacy enterprises, the digital settlers.

- In this book, you will learn some of the most illustrative success stories of the digital age, including digital settlers that have successfully transformed.

- You will learn the common methods used by successful organizations: unbundling digital business capabilities, designing digital business models through value exchange, and establishing widespread feedback loops.
- And you will see how those methods can be applied in specific strategies for your business, from optimizing digital channels to amalgamating value streams.

In all these areas, APIs feature prominently.

Treasure-Hunting Methods

The management and strategic practices of the twentieth century can only go so far in the digital economy of the twenty-first century. The most successful digital companies are utilizing methods tailored to the unique challenges and opportunities of the digital age.

Create Optionality Through Unbundling

The digital economy moves at breakneck speed. Innovation happens across all industries and all technical areas. New techniques are combined, leading to further innovations. The result is a business landscape that is as unpredictable as it is full of opportunities. Successful digital companies handle this pace by spending less time trying to predict the future and more time preparing for any eventuality.

Preparation comes in the form of optionality: unbundling their business capabilities into digital assets that can be combined and composed into new products, processes, and experiences that meet opportunities unlocked by innovation. These digital pirates accomplish this by creating APIs for reusable software functions and data that may be used in many different contexts. They also make these APIs as easy to discover and use as possible. This book will show you how to drive optionality in your business using APIs.

Identify Opportunities Through Value Dynamics

Business can be viewed on a macro level as a series of value exchanges (Clayton Christensen's term for the interconnected set of stakeholders that deliver value-producing products and services to a discrete set of customers[2]) between entities.

A small business might exchange its services for money from a customer. A wholesaler might discount its products to gain access to a market provided by a retailer. Digital business—with its penchant for high-speed, low-cost communication between organizations—reduces the cost of value exchange, allowing for more complex business models. Over time, value networks become increasingly complex, while the stakeholders within these value networks become increasingly optimized.

The unfolding of the web ecosystem has borne this out. APIs have been a frequent channel for value exchange by powering business-to-business (B2B) process integration, underpinning app-based commerce, monetizing data, and more. This book will teach you how to use value dynamics to identify opportunities for growth and innovation in your digital business. *Value dynamics* is a visual method of designing business models through value networks and API-enabled value exchange. It will help you "place bets" on opportunities that arise from optionality.

Optimize Value Through Feedback Loops

The rapid communication that unlocks more complex value networks also allows organizations to gain a continuous understanding of their business. When an organization is exercising many options simultaneously, feedback loops are critical for knowing as quickly as possible which options are succeeding and which need to be shut down.

When organizations first create APIs, they often look at them as a means of providing data to consumers. However, APIs capture data as well. Leading digital businesses use APIs to collect feedback on business activities that can guide digital strategy. This book will show you how to cultivate effective feedback loops that can help determine when to double down on option bets and when to fold.

In combination, these techniques require a change in mindset that may be uncomfortable for many organizations. Rather than defining a detailed, long-term plan, successful digital organizations are much more focused on creating conditions for opportunities to flourish and capitalizing when they do. They recognize that each successful innovation will lead to unanticipated opportunities, so they aim to deliberately harvest unintended positive outcomes—happy accidents. This book teaches you to remember your "OOOps"—the winning methods of *optionality*, *opportunism*, and *optimization*—if you want to benefit from happy accidents.

Successful Treasure-Hunting Strategies

By analyzing digital success stories through the lens of the three winning OOOps techniques, this book introduces four winning strategies that any business can employ to thrive in digital business using APIs.

TABLE 0.1: Four Winning Strategies in the Digital Economy

Exchange Optimization	Adapt analog business models for the digital world, thus increasing the speed and scale of value exchanges while lowering coordination costs.
Distributed Innovation	Put innovative technologies in the hands of business users and high option products in the hands of customers to run rapid, well-measured experiments that pay off.
Capability Capitalization	Unbundle and rebundle business capabilities to create new products, services, and even highly profitable operating units.
Value Aggregation	Connect disparate value networks to create an integrated business model whose whole is greater than the sum of its parts.

This book will help you use each of these strategies in any industry to great effect. We will also show you how to combine the winning methods to discover your own strategies. The sky's the limit!

We've Seen This Approach Work

Throughout our professional careers, we have both helped organizations bring their business and technology teams closer together in the pursuit of digital success. In the course of our work with enterprises, we've seen the same patterns and antipatterns emerge. From the private to public sector and across every industry, we have found that the recipe for success in the digital economy comes from the unbundling approach we share here.

Through our collective work, we have extracted and articulated the practices companies employ to build successful digital businesses. We codified these strategies and methods and then helped other organizations—especially digital settlers—apply them and yield positive results. These engagements inspired us to write this book.

In our research, we interviewed more than twenty individuals from a variety of innovative companies and across a wide spectrum of roles: CEOs, CIOs, architects, consultants, and more. You will hear firsthand from these change agents. Although they differ in roles, you will see how they share a common worldview:

- They see APIs as mission-critical business assets, not as technical components.
- They have an "outside-in" view of their business, orienting on their value proposition for customers more than the operational competencies their enterprises are known for.
- They weigh their near-term opportunities against a high-level mission rather than trying to adhere to a rigid, long-term plan.

The elements of this shared mindset show up in the strategies and methods of unbundling.

You Have a Part to Play

If you are a business or technology professional involved in your company's digital transformation, this book can help you. The approach we prescribe will especially help you align your organization's transformation efforts with meaningful business objectives. The strategies and methods articulated here will bridge the gap between the digital intentions of the C-suite and the day-to-day work happening on the ground.

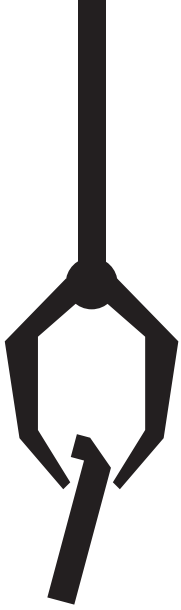
This book gives you a stake in your organization's transformation, whether you identify as a business leader or a technology professional. This is a book about connection: Connecting business to technology. Connecting long-term strategy to short-term return on investment. Connecting intuition to science. Anyone interested in helping their enterprise stay relevant in turbulent times will find value in this book.

Seeing It Through

To ultimately realize success, you will also need to confront some practical considerations. What are the organizational implications of taking this digitally native approach to business? What new risks are introduced in such a setup and how can they be mitigated? How can the approach be aligned with financial realities? This book answers all those questions.

For too long, business leaders have been trying to find their way in the digital economy through gut feel or by copycatting the surface characteristics of successful companies. This book goes beneath the surface, deconstructing those success stories into their essential lessons. This book shows that although the specific events or innovations may not have been intended, they occurred in environments that had been deliberately populated by API-enabled options. *Unbundling the Enterprise* defines a methodology for engineering those positive, unintended occurrences: a science of happy accidents.

Let's begin.



INNOVATION BY ACCIDENT

The chapters within Part I of this book explore the methods digital pirates like Amazon, Google, and Facebook used to fuel their explosive growth and sustain success over the last few decades. A curious pattern in these success stories is the prevalence of unintended events and innovations leading to some of their most positive outcomes. These "happy accidents" occur too frequently to be dismissed as coincidences. The focus of Part I is on discovering a methodology for intentionally engineering these unintentional outcomes, i.e., defining a science of happy accidents.

We will look at the common patterns adopted by digital pirates. To start, we will examine how these organizations have used APIs as foundational enablers for finding digital treasure. Through that examination, we will focus on three methods that lead to beneficial business conditions:

- Unbundling the digital landscape into API-enabled software components that provide discrete business capabilities, yielding a high degree of **optionality** within an organization's business environment.
- Rethinking business models through the dynamics of value exchange, thus providing a higher probability set of new business **opportunities**.
- Establishing feedback loops and reducing cycle time as a means of **optimizing** the digital operating model.

Each of these methods is explored in detail. Furthermore, we will show how digital pirates are not the only ones benefiting from these methods and the resulting happy accidents. We will also share stories of digital settlers who are finding success. It is time to set sail on the digital seas...

1

Treasure in Transformation

In 1882, Thomas Edison opened the first electrical power station in New York City.[1] To demonstrate the possibilities of continuous electrical power, Edison grabbed headlines by lighting up Lower Manhattan with electric bulbs. To everyone present, it was clear that the future had arrived.

But how could giant corporations that had emerged during the Industrial Revolution take advantage of this new power source? Their primary power source was steam, and they had spent the previous decades orienting their factories around steam boilers, optimizing the efficiency of steam power through new innovations, and creating a whole economy around steam-based products and occupations. Moving to a new power source would involve a lot more than unplugging steam and plugging in electricity. By 1900, only 5% of mechanical power was being provided by electricity.[2]

The tipping point for electrical power came from a new, electrically native industry. In 1913—thirty-one years after Edison's first power station—Ford Motor Company introduced the first moving assembly line for mass automobile production. This approach required discrete sources of power at multiple points along the process, something not possible with steam. By introducing a revolutionary manufacturing innovation that fueled the high-growth, job-producing automobile industry, Ford's assembly line helped unleash the electrical age.

Without being encumbered by antiquated approaches, the nascent automotive industry established many of the best practices for business in the twentieth century. Interchangeable parts, the division of labor, new management methodologies, and wholly controlled supply chains all became part of a business paradigm that spanned all manufacturing sectors. The model worked for the era. But those same practices that worked in the twentieth century don't seem to be as effective in the twenty-first century.

Today's period of digital transformation mirrors the shift from the steam age to the electrical age. It has been just over thirty years since the World Wide Web went online, the same amount of time it took for industry to transition from steam to electrical power. Companies formed within the last thirty years—Google, Amazon, Facebook—have grown to become some of the biggest in the world. These digital pirates have created new strategies and practices that fit the new age of business.

At the same time, many pre-web companies have struggled to optimize for digital business. Like early twentieth-century companies that adopted practices of the automotive industry, these digital settlers can learn from the digital pirates.

In this chapter, we will examine some of the common tools and practices digital pirates have used to thrive in the digital age. APIs are a factor, but it is how these companies use APIs that have led them to treasure. We will start with a source as surprising as a rolled-up piece of paper hidden in a pirate's boot.

The Yegge Platform Rant

One of the most detailed accounts of the digital pirates' inner workings was never intended to go public. In late 2011, Facebook was experiencing a meteoric rise and stood as the biggest threat to Google's dominance in web advertising revenue. Google's response was to launch the Google Plus social network. Google engineer Steve Yegge, who had previously worked at Amazon, posted a comparison of both Amazon and Google to the new social network. He thought it was being posted to a Google-only group, but it went

out to the public at large. According to Yegge's own account, he drank some wine, published the post, went to bed, and awoke to find he was on the front page of the *Wall Street Journal*. The post had gone viral due to its sardonic candor. But what makes it continually relevant is its articulation of Amazon's platform approach.[3]

Yegge, whose background included a great deal of work building tools for developers, dedicates much of his rant to the distinction between the platform approach he experienced when working at Amazon and the non-platform approach he was experiencing at Google. He was especially frustrated that Google Plus lacked what he saw as key platform capabilities.

In Yegge's view, a platform is a software-based system whose functionality is broken into discrete capabilities that are made accessible to developers through APIs. Whereas a product strategy is about trying to anticipate market and user needs and then delivering packaged products to meet those needs, a platform strategy is about giving the right tools to developers to test their own ideas and build their own products and services. "Platforms are all about long-term thinking," he remarks.[4]

In other words, by giving this collection of building blocks to developers, providing the right incentives, and aligning the right business model, platform owners can significantly diversify their strategic options over the long term. Yegge saw this as a key to beat rival Facebook's success at that time. "Facebook is successful because they built an entire constellation of products by allowing other people to do the work," he notes.[5] (This approach of "distributing innovation" across its digital ecosystem is examined more in Chapter 7.)

Yegge lived through Amazon's transition from a product company to a platform company. Through that process, he observed how a company the size of Amazon was able to make it work. "The Golden Rule of Platforms is that you 'Eat Your Own Dogfood,'" he notes. "'Eat Your Own Dogfood' can be rephrased as 'Start with a Platform, and Then Use it for Everything.'"[6]

Yegge credits Jeff Bezos directly for establishing the platform culture at Amazon. "Bezos realized long before the vast majority of Amazonians that Amazon needs to be a platform," Yegge claims. "[He] realized that he didn't need to be a Steve Jobs in order to provide everyone with the right products…

He just needed to enable third-party developers to do it, and it would happen automatically."[7] It was a strategy that distributed innovation among third-party entities.

Around 2002, Bezos sent a memo to all Amazon employees, according to Yegge's account. The memo—now often referred to as the "Bezos API mandate"—was directed at Amazon's software development teams. It stated that all data and functionality must always and only be exposed via network-based interfaces (such as APIs) designed to be used by developers in the world outside Amazon, and these APIs must be the only interaction point between teams internally. Given that the "Dread Pirate Bezos," as Yegge refers to him, generally followed up his company edicts with ruthless governance, the organization took note and the platform culture took hold.[8]

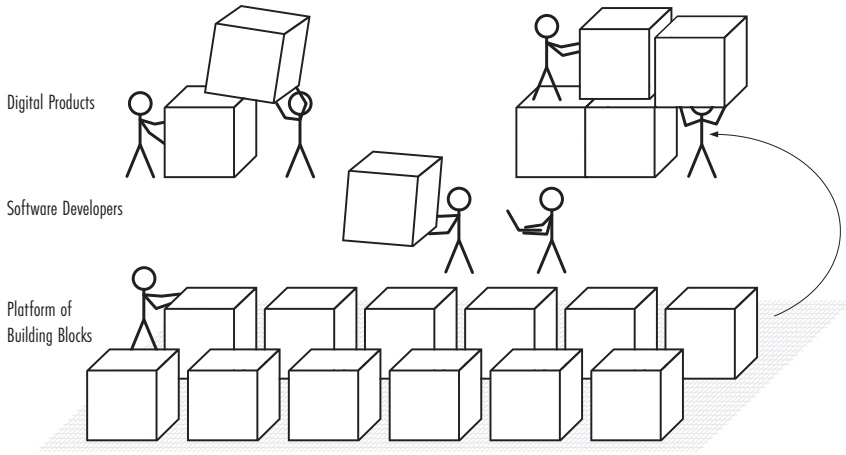


FIGURE 1.1: A Simplified View of the "Platform Approach" as Described by Steve Yegge

As Yegge recalls, "From the time Bezos issued his edict through the time I left, Amazon had transformed culturally into a company that thinks about everything in a services-first fashion."[9]

Amazon's platform culture led to a stunning run of product innovation. The API-enabled building blocks resulting from the platformization of Amazon.com were used to create offerings in new business domains, like

third-party marketplaces, payments, and authentication, and as the foundation for the Kindle. The most conspicuous innovation enabled by the Amazon platform approach was the launch of Amazon Web Services, which now accounts for a significant chunk of the company's revenue and profits. This was the most obvious example of enabling third-party developers to build on the Amazon platform. It is likely no coincidence that at the same time Amazon adopted its platform culture, the company also expanded its focus beyond the original vision of being the "everything store" and being much more open-minded and opportunistic about entering any adjacent market it could move to quickly:[10]

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another teams' data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
4. It doesn't matter what technology is used: HTTP, Corba, Pubsub, custom protocols.
5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
6. Anyone who doesn't do this will be fired. Thank you; have a nice day!

Yegge is one subjective source for this comparison of Google versus Facebook and Amazon, but the objective evidence supports his claims. The lesson here is that the platform approach adopted by Facebook and Amazon, and characterized using APIs, worked. The more closed, API-less approach Google used at the launch of Google Plus didn't.

Google would succeed with other products, which will be covered shortly. But first, let's look at Tim O'Reilly, one of the people who influenced Bezos's embrace of APIs.

Tim O'Reilly: The API Advocate

As early as 2000, Tim O'Reilly was enthusiastic about the possibility of the web as not just a collection of websites but as a platform of programmable capabilities that could be composed into novel services and embedded into new user experiences and devices. "Don't think of the web as a client-server system that simply delivers web pages to web servers," he remarked in his JavaOne Conference keynote in June 2000. "Think of it as a distributed services architecture, with the URL as a first generation 'API' for calling those services."[11] He specifically cited Amazon, eBay, E*Trade, and AOL as companies with data and functionality that could be unbundled and API-enabled to create this vast web platform.

Over the next two years, O'Reilly went on a personal mission to sell these companies and other web pioneers on his web-as-a-programmable-platform vision. AOL was unmoved by O'Reilly's pleas for it to open its recent MapQuest acquisition via APIs. Google responded conservatively by releasing their search function as an open API. According to O'Reilly, the most receptive company was Amazon. He was successful in getting an audience directly with the CEO. O'Reilly got Bezos's attention with an outside-the-box statement: "Amazon isn't just an e-commerce-site. It's become the information hub of the publishing industry."[12] Still, Bezos challenged O'Reilly to show how APIs could benefit Amazon. (Remember, this was two years before Bezos's API mandate.)

O'Reilly listed four reasons Amazon needed to adopt a platform culture by unbundling their capabilities through APIs. First, he used Microsoft as an example company that had repeatedly boxed out upstart competitors through their platform strategy. "Once you have other companies building added value that relies on you, you have a kind of benign industry lock in that's a real competitive advantage."[13] Here he predicted that MapQuest was

likely to be marginalized by Microsoft's MapPoint service for this "product versus platform" reason. Second, O'Reilly noted that APIs would provide a gateway for all the smart and creative people who work outside of Amazon to create unanticipated innovation for the company. As O'Reilly put it, "Giving developers a playground extends your development staff, bringing in new ideas and features at the same time as it builds your brand and image."[14] The third area of value that O'Reilly articulated to Bezos was the claim that opening APIs would lead to new revenue opportunities for Amazon if they were willing to be patient. "Disruptive innovations often don't work all that well at first," O'Reilly pointed out. "So, you have to give them room to grow before you try to harvest them."[15] He cited Google's ability to offer free service at low volume and paid services at scale as proof. O'Reilly's final argument to Bezos was that the web functioned like an ecosystem: "The more life there is, the more there is for everyone."[16]

Bezos was clearly moved by O'Reilly's appeal. He followed up with O'Reilly shortly after the meeting to let him know that Amazon already had some web-based APIs in development. Bezos even brought some of his key Amazon leaders along when he attended O'Reilly's Emerging Technology Conference in April 2002, a conference that focused overwhelmingly on the possibilities of the web-as-a-programmable-platform. It lines up that Bezos issued his API mandate shortly after these encounters with O'Reilly.

There were internal forces driving API adoption at Amazon as well. In a 2020 interview with Harvard Business School, Andy Jassy—successor to Jeff Bezos as Amazon CEO—recounted how Amazon was able to unleash parallel innovation within the company. Amazon had initially instituted a "New Process Initiative" to prioritize funding for new ideas. However, this process became a bottleneck and a hindrance to innovation. In Jassy's words, "We'd have twenty good ideas and only be able to fund two."[17] They were able to unstick the process by addressing their software architecture. "Our software was too coupled," remembers Jassy. "It was really hard to manage."[18] Amazon's teams began unbundling their business capabilities into "well-documented, hardened Application Programming Interfaces"[19] to remove the coordination costs that were slowing down their projects. "That took us several years to address [….] getting all the teams to have

these hard, well-documented APIs," says Jassy. "But that allowed us to, to kill the NPI (New Product Introduction) process so we could move it at a much faster rate."[20]

The constraints Amazon put in place to establish its platform culture had even more positive impacts than what O'Reilly had laid out. By forcing Amazon's own teams to communicate only through APIs, Amazon created a platform for external developers to innovate, setting the stage for developer-driven happy accidents to take place inside Amazon's walls. The mandate ensured that all of Amazon's software-based capabilities could easily be used internally to drive productivity and innovation, and only Amazon would decide which ones went public. That left room to keep select capabilities private to guard competitive advantage. Last, making every software service available to Amazon's developers lifted all of Amazon's boats, regardless of whether the tide rose for anyone else. Amazon had taken the platform approach to unforeseen heights.

Finding the Value of Google Maps

Picking back up on O'Reilly's vision for the future, it turned out that his warnings about MapQuest's lack of openness proved prophetic. However, it wasn't Microsoft's MapPoint that toppled MapQuest from its dominant position. Following Yegge's lamentation about Google Plus's lack of platform capabilities, he did tip his hat to a few Google products that were being designed and built with a platform mentality, even if it was against the current of Google's culture. One of those, Google Maps, provides even more insight into how packaged products can be unbundled into discrete capabilities that enable unanticipated innovations.

Launching as a product in early 2005, Google Maps aimed to dominate the online mapping market through its dynamic user experience, which contrasted favorably with market leader MapQuest's rigid UI. To enable the dynamic user experience, Google employed a new web architecture nicknamed AJAX (Asynchronous JavaScript and XML) that allowed client applications to send and retrieve data over the web without interrupting the

user. This architecture required Google to host a Maps API on the server side that facilitated the preloading of data on the client side, allowing users to easily pan and zoom on their maps. Google Maps was lauded as a revolutionary web application, and its market presence began to grow.

Along the way, some savvy developers who didn't work at Google figured out how Maps was built and thought of new ways to use the technology. Rather than connecting to Maps through the UI, they sought to use the raw data from the server side to create visualizations in their own applications. To do this, they connected directly to the API that underpinned the Google Maps application. Sites like HousingMaps.com and ChicagoCrime.org were early adopters of this approach.

When Google recognized what was happening, they faced a dilemma. Technically, these developers were breaking the Google Maps terms and conditions. However, by integrating Google's mapping data with other information on the web, they were completely aligned with Google's mission "to organize the world's information and make it universally accessible and useful."[21] Ultimately, Google embraced what the developers were doing. In June of 2005, then Google Maps Product Manager Bret Taylor announced the launch of the Google Maps API as its own product in a brief blog post titled "The world is your JavaScript-enabled oyster."[22]

Although they shared a common data backbone, the Maps web app and Maps web API were different products, and they evolved differently. By allowing sites to embed mapping data—thus bringing Google Maps to the users instead of making them come to Google—the Maps API was instrumental in Google's initial goal of unseating MapQuest. However, up to that point, the product had been effectively "going with the flow." According to former Google Maps API Product Manager Thor Mitchell, "At first, the long-term value of the Maps API to Google, or how it could generate revenue, was not questioned….It was more like 'this seems like this is in line with our goals, so let's support it.'"[23]

In June 2006, a year after the initial launch, Google introduced a paid version of the Maps API, but that was less about establishing a new revenue stream and more about appeasing corporate clients who would not trust a free service. The blog post announcing the new Google Maps for Enterprise

even states the new offering was created in "response to this popular corporate demand."[24] The potential value of the Google Maps API was apparent.

The mobile app boom that followed the launch of Apple's App Store in 2008 provided more opportunities to drive value through the Google Maps API. Its organic growth had been so steady that by 2011 the number of maps served by the Maps API exceeded the number served by the Maps web application.[25]

This got the attention of Google executives, and they worked with Mitchell and the team to solidify the Maps business model. First, as per its terms of service, Maps API usage could only be free for apps and sites that were free to users and universally accessible, in keeping with Google's mission. Second, limits were imposed to ensure the service was not being abused by consumers. This helped with cost recovery and avoidance, since half of the Maps API traffic came from a small set of highly trafficked websites, but it also led to some pushback from the customer community. Third, Google recognized that geolocation data would be central to its overall mobile strategy and worked to combine Maps and Places (check-in app) data to provide new personalized mobile experiences, as well as providing more precision in their targeted advertising business. By 2012, when Mitchell left his position as Maps API product manager, the number of apps and sites using the product had grown to more than one million.[26]

Google proceeded with the Maps product because it aligned with their corporate mission. The company observed within a few months of launching that customers wanted direct access to its Maps API, and Google was able to launch an API product only a couple months after that. In line with Tim O'Reilly's guidance to Jeff Bezos, the Google Maps team was "following its nose" as opposed to aiming for a fully realized product strategy. The Maps API kept Google's options open, allowing the company to act decisively as new opportunities arose. As a result, they gained first mover advantage. Later, they were able to react quickly to market dynamics—especially the mobile boom—to exploit the value of their Maps API product. "You make your own luck," as the expression goes. Serendipity in this case was only possible because Google had created the API option ahead of time.

H-API Accidents?

APIs played a big role in the growth and success of both Amazon and Google Maps. Amazon benefited from a top-down mandate that led to all its business capabilities being composable in new business lines and even business models. Google Maps took a more organic approach, following the wishes of its developer community to open via API and then riding the mobile boom to mass adoption. Neither company drew up a prescriptive five-year plan. So, are APIs the answer to success in the digital economy?

The good news is there are now many transformation success stories from companies that are digital settlers. Unsurprisingly, many of these have APIs as a common factor. Cox Automotive uses APIs to power its industry-leading marketplaces, such as Autotrader and Kelley Blue Book. Coca-Cola uses APIs to introduce new technologies across all its business lines and to crowdsource its product innovations. Capital One's APIs dynamically connect its customer channels with its core banking capabilities. Best Buy uses APIs to synthesize disparate lines of business. We explore each of these stories in later chapters.

As presented in the introduction, digital pirates have used APIs to find various forms of treasure: a viable business model for Facebook, a high-growth revenue stream for Google, the fulfillment of a grand vision for Amazon. However, this success did not just stem from the use of APIs. As we will see in the next chapter, these companies used three consistent methods to achieve these results. Those methods add up to an approach for uncovering happy accidents with APIs as the shovel.

2

The Science of Happy Accidents

Legend has it that pharmacist John Pemberton created the secret formula for Coca-Cola by accident.[1] The story goes that while working on an alcohol-based painkiller using coca leaves and kola nuts, he decided to remove the alcohol in response to the temperance movement that was gaining popularity at the time. As the tale tells, he accidentally mixed sugar syrup and carbonated water with some of the original ingredients, leading to what would become the most well-known soft drink in the world. Whether this story is true or not, there have been many substantiated happy accidents like it in every era of business.

In 1966, Richard M. Schulze opened a new electronics store in his hometown of St. Paul, Minnesota. He named it Sound of Music to capitalize on the previous year's most popular movie. After growing the business modestly for fifteen years, it would be another classic film—*The Wizard of Oz*—that would prove a better analogy for the company's fate. In 1981, a tornado tore the roof off the biggest Sound of Music location. To make as much revenue as possible from the store's surviving stock, Schulze promoted a huge "Tornado Sale" featuring "best buys" on all remaining merchandise. The sale was a massive success and set the stage for the company to change its name to Best Buy, as well as change its go-to-market strategy to focus on low prices and high volume.[2]

After selling Flickr to Yahoo! in 2005, founder Stewart Butterfield started a game company called Tiny Speck that had some modest success with the multiplayer online game *Glitch*. Although that game failed to reach Butterfield's expectations, he recognized that the collaboration capabilities offered by the game had singular potential. To capitalize on this opportunity, Butterfield morphed Tiny Speck into a new company, Slack, and took aim at an entirely new class of customers with its new stand-alone collaboration product.[3]

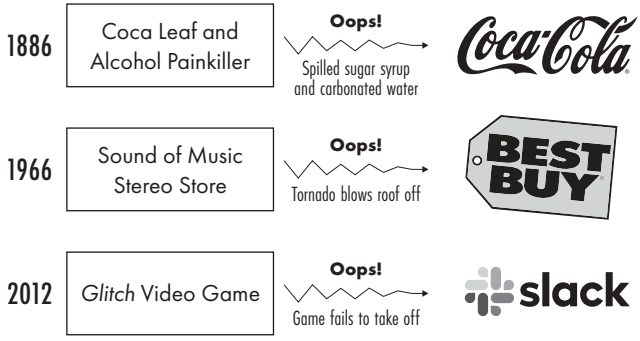


FIGURE 2.1: Happy Accidents Through the Years

In each of these situations (see Figure 2.1), an unsuccessful invention led to an unintended innovation. As the Slack example shows, the digital age has happy accidents too. However, as we have seen, the examples of Google Maps and Amazon Web Services seem to be less accidental than the legend of the Coca-Cola formula. Amazon and Google had strategies that enabled them to exploit new opportunities that arose from the combination of changing conditions in the marketplace and the capabilities they had created.

Combinatorial Innovation

There is a popular saying in the tech world that we overestimate the change that will happen in one year and underestimate the change that will happen in ten years.[4] This phenomenon is explainable. When people predict the future of

technology, they often look at things linearly. For example, to predict the future of commuting to work, people might focus on the evolution of cars. This might lead them to exaggerate the projected adoption of self-driving cars. However, there are many interconnected conditions that could affect the future of how people get to work: alternative transportation mechanisms, the ratio of remote to in-person work, combined urban zoning, even the definition of work itself. It is likely that the concept of commuting will greatly evolve in the next decade, and that evolution will be affected by all the areas mentioned.

We underestimate longer-term change because we fail to see that innovation happens in combination. New innovations are built on a synthesized foundation of multidisciplinary capabilities. Consider one of the most impactful new technology paradigms of the twenty-first century: the mobile phone explosion circa 2008. The iPhone was not the first smartphone, but it was the innovation that introduced the new mobile paradigm. The iPhone itself contained some impressive hardware, but there were many conditions that led to its profound success that were external to the device.

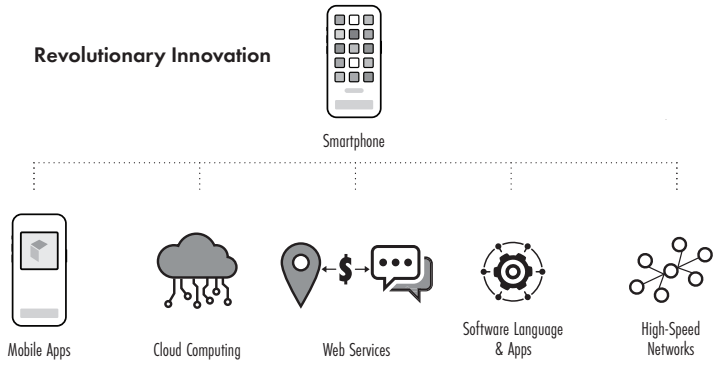


FIGURE 2.2: The Smartphone Innovation Was Enabled by Concurrent Capabilities

Here is a list of just some of the capabilities that allowed the mobile paradigm to take hold (see also Figure 2.2):

- Mobile apps and a mature World Wide Web provided engaging services that incentivized habitual usage of the new phones.

- Cloud computing provided highly available application infrastructure with elastic scalability and ubiquitous access.
- Web APIs provided a familiar way for app developers to connect to their own backend data as well as to fundamental third-party services such as telephony, payments, and geolocation.
- Mobile-friendly software languages and frameworks allowed highly motivated developers to ramp up quickly enough to launch their apps in good time.
- High-speed cellular networks provided the capability of piping the new volumes of mobile data.

A mobile phone expert from 1998 may have predicted touchscreen keyboards and high-resolution video playback, but it is unlikely they would have foreseen the App Store or the asymmetric treasure it delivered to Apple.

These examples illustrate a couple of points. First, the biggest innovations—the ones that offer the biggest business opportunities—tend to stem from a combination of innovations that happen in multiple related fields. Second, we have no way of knowing what those big innovations will be. If we accept these two statements, then we can see what's wrong with trying to build a five-year plan. Instead, organizations should focus on cultivating an environment in which happy accidents can take place.

As articulated above, innovation comes from a combination of capabilities. It makes sense then to propose two reasonable hypotheses when it comes to digital strategy: (1) organizations should strive to have a lot of capabilities, and (2) those capabilities need to be "combinable" or "composable."

The mobile paradigm validates these hypotheses on a grand scale. Amazon shows the hypotheses to be true on the level of a single organization. The Bezos mandate and its resulting institution at Amazon led to the creation of a tremendous number of capabilities. The mandate also ensured those capabilities were built in a composable way, as each capability could be accessed via API and therefore utilized within many processes and across many digital channels.

Both Google Maps and Slack show how specific capabilities can be made composable. Slack had a capability—multi-user collaboration and communication—built into its gaming context. This capability needed to be unbundled

from the game to become composable. Once unbundled, the capability was turned into a product that served a much more generalized market.

From a technology perspective, Google Maps had an API for its UI-based product. However, it wasn't until third-party app developers reverse engineered the API that Google turned its API into a formal product. As described previously, that product became foundational to the mobile boom and provided new revenue and a new channel for Google's business.

These digital pirates didn't use a treasure map to find treasure. But they did use a consistent methodology, even if they didn't realize it at the time.

The Non-Rival Nature of Digital Products

Besides combinatorial innovation, there is another dynamic at play in the digital economy that accelerates the hurried pace of change. In a manufacturing economy, there are constraints that dictate the speed and cost of sourcing, storing, combining, and distributing physical goods. During the manufacturing boom of the twentieth century, companies like Ford would gain a competitive advantage by controlling their entire supply chain to minimize the impacts and unpredictability of those constraints. Digital products are different.

In economic terms, data is a *non-rival good*. A rival good is one that can only be consumed or possessed by one party at a time. Physical products are rivalrous. Two people cannot drive the same car to two different places at the same time. Non-rival goods, however, can be consumed or possessed by multiple parties simultaneously. People can stream the same songs and movies all at once. An original Bob Ross painting is a rival good, whereas a broadcasted Bob Ross television show is a non-rival good.

Rival products are limited by unit costs. The economic return on producing rival products is constrained by the cost of producing each item. Furthermore, their ultimate return is often capped by logistic constraints within their market. This leads to a *concave curv*e when plotting their costs versus returns. Non-rival goods have the opposite property. Because each unit can reach an unlimited number of consumers, their potential returns

are unconstrained by unit costs. This means they have a *convex curve* when plotting their potential returns versus production costs (see also Figure 2.3).

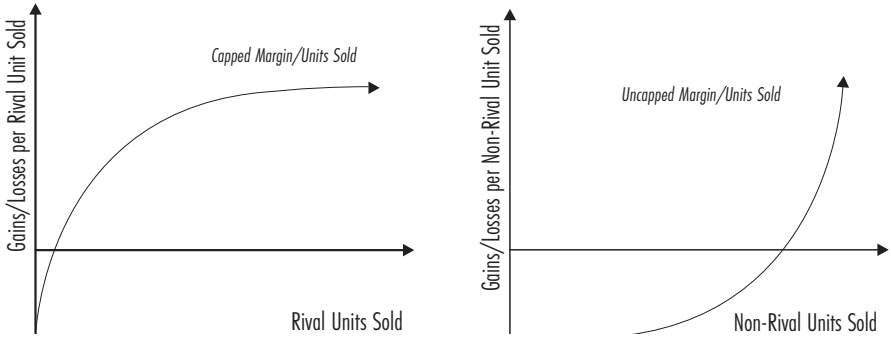


FIGURE 2.3: Concave vs. Convex Margin Curves of Rival and Non-Rival Products

Digital technologies make it easy to move data at speed and scale. They make it easy to store data cheaply. They make it easy to combine and transform data. However, it is also this economic property of data—its non-rival nature and subsequent replicability—that multiplies its economic potential. APIs are digital products whose material is data, making them non-rival products. Therefore, organizations launching successful API products create this potential for convex returns on their investment.

When it comes to returns on digital products, the degree of convexity is driven by the cost of experimentation. The cheaper it is for an organization to create, launch, and manage digital products, the steeper the curve on their returns. That's not to say that organizations should just launch as many digital products as possible. Instead, it's about how easily opportunities can be identified, how quickly the products can be assembled, and how readily their success or failure can be determined.

Digital pirates have used APIs to cheapen experiments. Having their business capabilities accessible via APIs accelerates the assembly of new products and services. Having these APIs reusable across channels and segments means they can easily be incorporated into new opportunities with minimal incremental cost. Digital pirates also use APIs to provide data on the performance of their experiments, which they use to increase returns by dou-

bling down on successful ones and quickly shutting down failed ones. Thus, with APIs, experiments become cheaper, creating even more fertile ground for happy accidents. Taking all of this into consideration, let's now lay out a methodology for applying these principles in a practical way.

OOOps: A Science of Happy Accidents

Using the examples we've already walked through as well as several more we will study, we propose a science of happy accidents consisting of three methods. First, you need to create *optionality* by unbundling business capabilities through APIs. These API-enabled business capabilities are like your mechanical pirates digging for treasure.

Second, you need to use value dynamics to identify the *opportunities* that represent the best starting point for your experiments. We will explore value dynamics in detail in subsequent chapters, but for now, you can think about it as a way of finding the right island for your pirates to excavate.

Last, you need to establish the right feedback loops to help you *optimize* your innovation experiments. This helps your pirates know where to stop digging and where to double down. Collectively, we call these three methods—optionality, opportunity, optimization—*OOOps*. (See also Figure 2.4.)

- Create **optionality** through unbundled APIs.
- Identify **opportunities** through value dynamics.
- Drive **optimization** through feedback loops.

FIGURE 2.4: OOOps: The Three Methods in the Science of Happy Accidents

Optionality Through Unbundling

As you approach your own life, do you make significant commitments as soon as possible or as late as possible? If you're seeking a new job, working with your children through the college application process, or even using an online dating app, do you commit to the first possible choice that comes your way or do you try to slow the decision down until more information becomes available? Slowing down the process of making commitments conserves optionality.

Using APIs to expose business capabilities allows an enterprise and its partners to slow down the process of making a commitment for exactly how a capability will be used, allowing the business to pursue multiple options at once. Software systems built using APIs allow optionality beyond the intended scope of an individual solution without significant sacrifice. In the case of Google Maps, Google never intended to turn its API into a paid product. However, because they chose not to lock themselves into a single context of use, businesses around the world were able to leverage the API to seek their own treasure. Rather than Google choosing to put a price tag on the API product, it was the users of the API who asked to pay for it to ensure that they could depend upon it as they incorporated Maps into their business models.

Optionality can be created in any software system. Consider a bank that wants to provide a self-service account open option for its retail customers. The bank would want this process to be as streamlined as possible, but at minimum, it would involve collecting customer information, checking credit scores, providing and vetting legal forms, giving choices for account types, and creating the new account, as well as tracking and communicating the status of the whole application. Figure 2.5 shows one way to solve it: build a monolithic web app that provides a self-service user interface and integrates with existing backend systems.

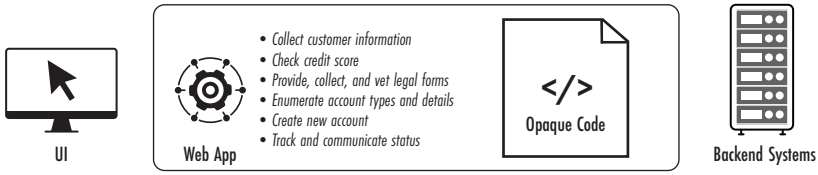


FIGURE 2.5: A Monolithic Web Application for Self-Service Account Open

This solution meets the stated requirements and that's it. But this same service could be delivered with a much higher degree of optionality. Figure 2.6 unbundles five capabilities from the web application and exposes them through APIs. The user experience remains the same, but now the bank has the option to take these capabilities out of the self-service account open context and use them in other ways. Customer information and retail account services can be leveraged at other customer service channels, such as in branches or at the call center. Document management and workflow tracking services can be used in numerous business processes. Credit scoring could conceivably be packaged and sold to third parties. Furthermore, the bank has more flexibility to make changes if this account open process needs to be augmented with new capabilities or if changes need to be made to its existing capabilities. The bank may not know what all the options are for the future, but it will be poised and ready through unbundling.

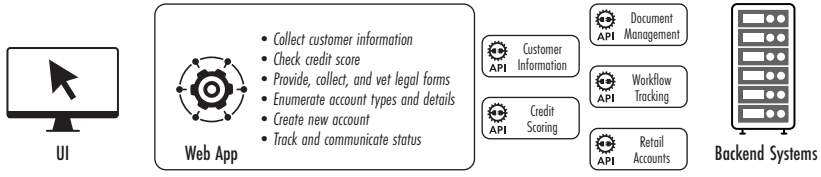


FIGURE 2.6: Self-Service Account Open Using Unbundled Capabilities via API

For decades, industry experts have advocated for modularity in software architecture, from early papers on information hiding through service-oriented architecture and microservices. However, choosing to make a decomposed architecture made up of discrete capabilities isn't just a technical choice. There are real financial benefits to creating decontextualized digital capabilities. Once unbundled in this way, these APIs can be rebundled into countless new contexts.

Opportunism Through Value Dynamics

Ultimately, the value of digital systems can be measured by how well they align with an organization's business model. Alexander Osterwalder, creator

of the Business Model Canvas, defines business models as the way companies create, deliver, and capture value.[5] We expand on that to define *value dynamics* as a visual method for mapping out business models by illustrating the flow of value in a digital ecosystem. These business model maps use a few simple elements:

- Shapes to depict constituents in the ecosystem, such as companies or consumers.
- Arrows to show the flow of value between constituents.
- Icons that indicate what type of value "currency" is being exchanged. There are some obvious value currencies and some that aren't as intuitive.

Figure 2.7 shows a basic example of value dynamics using the relationship between a retailer's business model with that of a wholesaler in the same customer ecosystem. Obviously, the retailer purchases products from the wholesaler, which it then sells to end customers. However, visualizing the business model allows for the articulation of the retailer's unique role in the ecosystem. The retailer can buy products for a lower price from the wholesaler, since it is also providing reach to a targeted set of customers aligned with the wholesaler's products. On the other side of the picture, the retailer can mark prices up to consumers, since it is providing time savings and convenience in the form of store locations, product curation, customer service, and more.

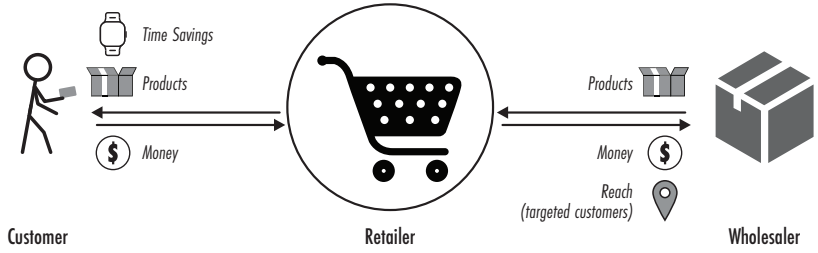


FIGURE 2.7: Simple Illustration of a Value Dynamics Map

Value dynamics allows you to illustrate a simple view of the value exchanges happening in your current digital ecosystem, thus depicting your business model according to the Osterwalder definition. Understanding how your business model works today, you can overlay what new capabilities or "options" you have; consider what value those capabilities capture, deliver, or create; and then see how that can flow between constituents in the ecosystem. From there, you can look for opportunities to add, augment, or eliminate value exchanges and even think through what new value can be captured on existing links. By using such a constrained set of items, the whole process can be gamified.

Optimization Through Feedback Loops

A common theme across the first two OOOps methods is lowering the cost of experimentation. Unbundling and optionality provide a faster and cheaper path to running experiments by providing digital building blocks. Value dynamics points the way to starting points that have a higher probability of yielding useful results. The third OOOps method—optimization—is about setting up feedback loops that help continuously decrease the cost of experiments and time to value.

Software movements such as Agile methodologies and DevOps stress the criticality of paving the path to learned insight. The notion of feedback loops and situational awareness are prevalent in any management methodology focused on dynamic environments. Back to the treasure hunt analogy, once those pirates start digging on the selected island, how do they determine when to stop digging or when to bring in more shovels? There are four main themes (see Figure 2.8) that emerge for how enterprises can ruthlessly lower the cost and time required for teams to create and test bundled and unbundled packages of value and deliver consumer-facing experimentation at scale.

Feature flags provide the ability to make targeted changes in API-enabled capabilities to controlled sets of audiences. From the color of a screen element to a change in the flow of screens for any user task to an alternate execution route via different APIs, feature flags can not only aid in optimizing the experiences of your developers and users but also help to control the risks and costs of your experiments.

Making it easy to segment traffic is the second tool to control the size of audience members receiving experimental changes. *Ramps* enable your experimentation teams to funnel a controllable percentage of all application traffic, again with simple configurations controlled via automation.

Just like a DevOps professional might use a waterfall chart to understand performance bottlenecks, your experiment teams will need robust *visualization* tools to make the efficacy of experiments self-revealing.

Finally, both *statistical literacy and tooling* are required to interpret the measurements derived from feedback loops. While the tooling capabilities are critical to making this approach work, it's equally critical to have a team of qualified humans to interpret the results. Both your business and technical teams need to be deeply familiar with the language and tools of statistics. Mature teams know that data beats math, and elite teams know that statistics tools can be misleading and cause speed-killing friction when you don't have the expertise to discern what the results mean and what they don't.

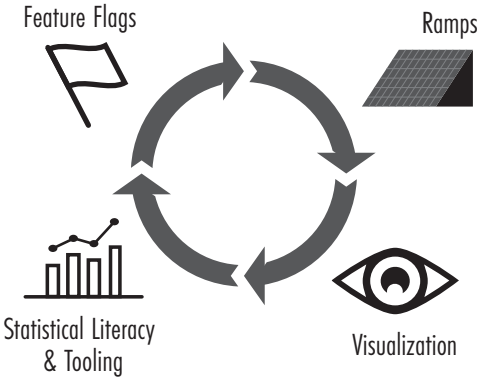


FIGURE 2.8: Four Feedback Mechanisms

In addition to these techniques, APIs create their own feedback loops. People commonly think of APIs as sources of data, a place where information can be retrieved or services can be requested. However, every API interaction creates contextual metadata that can be leveraged for other sources of value. Look at the example set by Google Maps. Maps consumers were happy to pay Google to get geolocation data in real time. But in so doing, those consumers were also providing information about their own locations and signaling

intent by the requests they made. When the mobile boom happened, Google was able to profit from this geolocation context by using it in ad targeting, a strategy we will explore more deeply in Chapter 9. We will see many other examples where APIs have been used to drive useful strategic and operational feedback loops.

Feedback loops provide the instrumentation in your digital ecosystem. These four feedback mechanisms, and even APIs themselves, allow you to continuously measure value exchange and value impact in your ecosystem. Doing this will help you drive down the cost of creating new options and drive up the return on investment of exercising those options.

The three OOOps methods work most effectively together. If you just created APIs out of every software function in your environment, you might have a lot of options, but they could be rife with duplication and misaligned with the value-generating areas of your business. Value dynamics on its own could provide some useful starting points for digital innovation, but those opportunities would be difficult to explore without options and be merely speculative without feedback mechanisms. Last, feedback without a strategic direction would just provide a more accurate means of demonstrating stagnation. Combining the three OOOps methods, however, yields remarkable results.

OOOps for Digital Settlers

The OOOps methods haven't only been proven by digital pirates. Progressive Insurance is one digital settler who has utilized API-enabled unbundling to drive innovation. True to its name, Progressive has a history of disrupting the consumer insurance industry.

In the early 2000s, at the same time Amazon and Google were first introducing their web APIs, Progressive was already experimenting with APIs of its own. In its Claims group, Progressive launched an application called "total loss concierge" that helped drivers obtain financing if their vehicles had to be written off following a claim. To power the application, "[Progressive] used SOAP (Simple Object Access Protocol) to connect directly with banks to get financing options," recalls Geoff McCormack, one of Progressive's enterprise architects who remains with the company today.[6] This early move turned the

company onto the value of unbundling through APIs, as well as the opportunism of digital ecosystems.

In the late 2000s, Progressive introduced Snapshot, a pluggable device that collected driving data used to assess drive safety. Customers could opt in to use this device to gain discounts on their auto insurance premiums. In the data-driven insurance industry, Progressive was able to disrupt the standard approach of pricing primarily on variables and instead utilize dynamic data. The company used an approach akin to the second OOOps method—opportunism through value dynamics—implicitly in this innovation. Furthermore, they were basing the innovation by optimizing through feedback loops (OOOps method #3) of driver information.

Progressive has also employed the first OOOps method: optionality through unbundling using APIs. For the last few years, Progressive has adopted an API-first approach to building its software systems that is paying dividends. Analyzing this unbundled landscape of business capabilities, Progressive recently launched a product targeted at small business customers that bundled its own APIs with a third-party, API-based insurtech provider of underwriting services. This brought the product to market in a matter of months, well ahead of expectations. "Our mindset is all around innovation, and we see the value in partnerships," McCormack states.[7] This ecosystem mentality is well-aligned with value dynamics.

Down to the Depths

Progressive Insurance is just one example of a company established prior to the web—a digital settler—figuring out the right approach to thriving in the digital economy. Although they weren't using a playbook for digital innovation, they adopted principles and practices used by the digital pirates to good effect. The next few chapters provide a playbook to leverage the learnings of digital success stories. We will go into each of the OOOps methods, illustrating how they work in action and how they combine to define a science of happy accidents.