

Simplify the Complexity
of Creating Digital Products



UNTRAPPING PRODUCT TEAMS



DAVID PEREIRA

Forewords by Ash Maurya and Jim Highsmith

FREE SAMPLE CHAPTER |



Praise for *Untrapping Product Teams*

“This is my new favorite book on product management. *Untrapping Product Teams* covers everything you need to know to lead a product team and be successful as a product manager. Author David Pereira does an excellent job of pointing out biases and thinking traps that doom products. The book is full of many insights and tools that will be useful for years to come.”

—Mike Cohn, Co-founder, Agile Alliance

“David’s book shares several hard-earned lessons of what happens when product leaders, product managers, and especially product owners are not trained to succeed in their jobs, and they go on to make predictable and avoidable mistakes. This book can help you avoid some of these pitfalls.”

—Marty Cagan, Partner, Silicon Valley Product Group

“This book touches on all the daily essentials for a product person. It’s a practical guide and a meta-analysis rolled into one, serving as the ‘Greatest Hits’ album of product management. Ideal for newcomers and an excellent refresher for those already immersed in the field.”

—Petra Wille, author of *Strong Product People*

“David’s brilliant book will help you decipher the chaos of product management and show how to escape all the traps that may condemn your product teams to mediocrity.”

—Maarten Dalmijn, author of *Driving Value with Sprint Goals*

“*Untrapping Product Teams* is the best way to transition your teams from delivery to empowered. Help your product teams break free of constraints and change the trajectory of your business with David’s book. He speaks not just theory, but from experience.”

—Aakash Gupta, author of *Product Growth*

“David wrote the book I wish I had read 20 years ago. Instead, I faced these common product team traps without a map of how to practically navigate them. This book is full of wisdom and inspiration for new and experienced product professionals alike. It will save you a lot of time and energy!”

—Simonetta Batteiger, CEO, Inclusive Leaders

“David’s book is an antidote to bullshit management, and at the same time, it’s so much more than that. It’s a wake-up call for product teams to regain clarity and focus. It’s a survival kit for every PM in the trenches facing a reality that other product books fail to acknowledge. There is no fluff, no ideal-world scenarios, only experience-based learnings and a generous amount of eye-opening honesty.”

—Ioana Ognibeni, Product Lead and Coach

“For those seeking no BS product management, David has got you covered! This book is full of practical and easy-to-follow advice to help you stop building a feature factory. David provides you simple frameworks and templates to use immediately.”

—Shyvee Shi, Product Lead at LinkedIn; author of *Reimagined: Building Products with Generative AI*

“Real-world product management is messy and full of traps that prevent great outcomes. Use David’s experience and nonprescriptive guidance to make good choices and help your company succeed.”

—Gil Broza, Value Delivery and Agile Leadership Expert;
author of *Deliver Better Results*

“I worry that many aspiring product managers will read Part I of this book and run away screaming. David lays bare the challenges that many will encounter in this role. The brave ones who venture to Part II will find practical advice and tools that will enable them to overcome those challenges and set them up for real success. If they press on to Part III, they will know how to stay that way. Plenty of experienced product people could also learn a thing or two to help them untrap themselves and their teams!”

—Phil Hornby, Product Coach, for product people

UNTRAPPING PRODUCT TEAMS

SIMPLIFY THE COMPLEXITY OF CREATING
DIGITAL PRODUCTS

DAVID PEREIRA

 Addison-Wesley

Hoboken, New Jersey

Cover images: yod 67/Shutterstock; Ihnatovich Maryia/Shutterstock.

Author photo: Courtesy of David Pereira. Photo by Anna Hermann.

Multiple figures created by the author with the help of Canva. Go to <https://www.canva.com/> to learn more.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Please contact us with concerns about any potential bias at pearson.com/report-bias.html.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2024933282

Copyright © 2024 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

ISBN-13: 978-0-13-533538-3

ISBN-10: 0-13-533538-8

\$PrintCode

To the most important people in my life:

Maria, my mom, who raised me with unconditional love.

José Antônio, my dad, who always encouraged me to follow my dreams.

Kevin, my brother and best friend, who has always been by my side no matter what.

Anastasia, my wife, who truly understands me and supports all my wild ideas.

This page intentionally left blank

Contents

- Foreword by Ash Maurya** **xiii**
- Foreword by Jim Highsmith** **xv**
- Preface** **xvii**
- About the Author** **xxi**

- Introduction** **1**
 - Who Can Benefit from This Book 2
 - How to Read This Book 3
 - What to Expect from *Untrapping Product Teams* 3

- Part I Facing Dangerous Traps** **5**
 - 1 How Common Ways of Working Trap or Untrap Teams** **7**
 - How Feature Factory Companies Work 8
 - Coordinative Flow: A Logical Way of Working with Unexpected Results 12
 - Nine Out of Ten Ideas Will Fail 15
 - Collaborative Flow: A Simple Way of Working with Outstanding Results 16
 - Key Takeaways 21
 - 2 The Mindset Impacts on Dangerous Traps** **23**
 - What’s a Growth Mindset? 25
 - What’s a Fixed Mindset? 25
 - Identifying the Mindset’s Impacts 26
 - Output over Outcome 27
 - Opinions over Evidence 28
 - Prescriptive Roadmaps over Embracing the Unknown 30
 - Meeting Deadlines over Achieving Goals 32
 - Implementing Solutions over Solving Problems 33
 - Key Takeaways 35

3	What Distracts Teams from Creating Value	37
	Facing the Real Enemy: Bullshit Management	38
	Gathering Requirements: A Distraction to Value Creation	39
	Extensive Product Backlog: When the Past Limits the Future	40
	Output Reports: Wasting Time	42
	Consensus-Driven Decisions: Slowing Down Progress	43
	Output Approval: Limiting Team's Accountability	44
	Meeting Marathon: Long Days with Little Value Created	45
	Fear of Saying No: The Art of Getting Overwhelmed	46
	Bridge Communication: Becoming the Bottleneck	47
	Opinion-Driven Decision: Ignoring Evidence, Creating Waste	48
	Failure Avoidance: Killing Innovation Chances	50
	Bullshit Management Check	51
	Key Takeaways	51
Part II	Overcoming Dangerous Traps	53
4	Using the Right Ingredients for Product Teams	55
	How a Simple Restaurant Became the World's Number One	56
	Understanding the Right Ingredients for Product Teams	56
	What a Solid Product Strategy Enables	57
	Accelerating Learning with Product Discovery	59
	Approaching Product Delivery Mindfully	61
	Finding the Right Ingredients for Your Situation	62
	Key Takeaways	63
5	Crafting a Meaningful Product Strategy	65
	Identifying Who You Serve	66
	Understanding Your Product Stage	68
	Product Vision: Knowing Where to Land	69
	Lean Canvas: Defining the Key Aspects to Thrive	72
	Value Curve: Designing to Differentiate	80
	Roadmap: Establishing Clarity on What to Achieve	82
	Key Takeaways	86

6	Forget What You Know to Get Ready for the Unknown	89
	My Unforgettable First Date with Product Discovery	90
	Biases: Blocking Learning Opportunities	91
	Confirmation Bias: The Art of Missing Opportunities and Creating Waste	92
	Commitment Escalation: Making a Bad Situation Even Worse	92
	Sunk-Cost Fallacy: The Fear of Losing Triggering More Losses	93
	Bandwagon Fallacy: Following the Masses Without Knowing Why	94
	Value Drivers: Doing What Matters for Customers and the Business	95
	Do What's Right, Not What's Easy	97
	Key Takeaways	98
7	Embarking on a Journey to Discover What Matters	99
	Understanding the Discovery Journey	99
	Clarify the Goal: Know What Success Looks Like	101
	Understand Your Audience: Know How Customers Use Your Product	102
	Uncover Value Drivers: Identify What's Worth Pursuing	104
	Prioritize and Focus: Take One Value Driver at a Time	107
	Reinvent the Future: Diverge and Converge	109
	Deal with Assumptions: Drop Bad Ideas Fast Enough	110
	Use Evidence to Progress: Experiment, Learn, Decide	116
	Key Takeaways	118
8	Enabling Product Delivery Beyond Outputs	119
	How the Wrong Approach to Product Delivery Leads to a Feature Factory	120
	Product Backlog: Organizing What Drives Value	121
	Refinement: Building Shared Understanding	125
	Alignment: Setting the Next Goal	127
	Getting Things Done	129
	Using Tech Debt: Accelerate Learning to Avoid Poor Investment	131
	Review: Getting Business and Product Aligned	134
	Improve: Strive to Continuously Become a Better Team	135
	Key Takeaways	136

9	Defining and Measuring Actionable Metrics	139
	Data Traps: Avoid Common Traps	140
	Leading versus Laggard Metrics	141
	Input versus Output Metrics: Understanding Control and Influence	143
	Defining Metrics: Focus on Signs While Ignoring Noise	144
	Measuring Results: Be Careful, You're Not a Stockbroker	146
	Sharing Results: Spread the News about Outcomes	147
	Pro Hints: Saving You Hours per Week	148
	Usual Metrics: The Bread and Butter of Analytics	149
	Key Takeaways	151
10	Getting the Teams Right	153
	Common Product Teams Setup	154
	Roles and Responsibilities: Who Takes Care of What	160
	Product Managers versus Backlog Managers	162
	Software Engineers versus Coders	165
	Product Designers versus Pixel-Perfect Designers	167
	Agile Coaches versus Agile Rulers	168
	Key Takeaways	169
11	Building Solid Relationships	171
	Who the Stakeholders Are	172
	Power and Interest: How to Improve Collaboration with Business Stakeholders	173
	Sharpening Communication: How to Ensure Shared Understanding	175
	Establishing Alignment: No More Confusion	177
	Breakthroughs Occur with Real Interactions Instead of Shallow Ones	179
	Key Takeaways	180
Part III	Remaining Untrapped	183
12	Establishing Product Principles	185
	Why Product Principles?	186
	How to Craft Product Principles	186

How to Ensure Principles Are Kept Alive	188
Evolving the Product Principles	190
Principles That Helped My Teams	190
Key Takeaways	196
13 Assessing Your Company's Dynamics	197
Leadership: Creating a Space Where Great Ideas Can Happen	198
Product Perception: Are You Ready to Thrive?	200
Product Leadership: Coordination or Collaboration?	201
Team Topology: Setting Teams Ready to Rock	202
Risk Tolerance: From Fear to Courage	203
Key Takeaways	204
14 Evaluating How Your Product Strategy Implementation Simplifies Decision Making	205
Ownership: Clear Accountability, Better Results	206
Prioritization: Moving from Serving to Partnering	207
Decision Making: Progress over Analysis	208
Goals: Clarity on Where You Want to Land	208
Collaboration: Getting Together to Create What Matters.	210
Key Takeaways	211
15 Understanding How Quickly You Can Drop Bad Ideas	213
Value Proposition: Clarity on What's Most Important.	214
Customer Interviews: Creating Insights to Unlock Learning	215
Identifying Assumptions: Knowing What We Don't Know	216
Testing Assumptions: Confronting Reality	217
Ideation: Creating a Better Future	218
Key Takeaways	219
16 Examining How Fast You Can Drive Business and Customer Value	221
Implementation: Prioritizing Learning over Scalability	222
Measure Results: Focusing on What Moves the Needle	223
Feature Removal: Getting Rid of Distractions.	223

Facing Failures: Learning versus Blaming	224
Product Backlog: Past Promises versus Future Opportunities	226
Key Takeaways	227
Wrap-Up	229
A Untrapping Lessons	233
Bibliography	235
Index	237

Foreword by Ash Maurya

Untrapping product teams. When I first ran into those words on David’s blog, I knew he had something unique to say. David has been on a mission to hone the art of product management—not simply to crank out more products, but to build products that customers truly love.

As products have gone from being delivered in a box to being delivered over the internet, there’s been a dramatic shift in how customers consume, demand, and interact with products. Customers no longer want to wait for long release cycles. They want value delivered to them continuously. This fundamentally changes how products need to be built.

It is no longer enough to simply throw more features at customers. That way of building products used to work at a time when there were huge barriers to entry and few competitors. Even if you got the product completely wrong, you had time to course correct and get back on track.

But fast forward to today. . . . With the internet, open source, and cloud computing, it has become cheaper and faster than ever to introduce new products, which means there is a lot more competition than before—both from incumbents and from new companies starting up all over the world.

In the old world, failing to deliver what customers wanted led to failed projects. But in the new world, continuously failing to deliver what customers want can lead to total business model failure. This is because customers today have many more choices than before. If they don’t get what they want from your product, they’ll simply switch to something else.

Delivering better customer outcomes in the new world requires rethinking traditional product planning, prioritization, and delivery processes. You can’t afford to spend long cycles analyzing, planning, and executing your idea. Speed of learning is key. You need a more iterative approach that replaces top-down planning with continuous discovery and validation.

This is the essence of continuous innovation.

When companies learn fast, they outlearn their competition and get to build what customers truly want. When you outlearn your competitors once, you stand ready to launch a new innovative product and capture market share. But whatever is worth copying will eventually be copied. Only by continuously outlearning your competition can you stay relevant to your customers and see your business model continuously thrive and grow.

The good news is that over the last decade, we’ve pushed the envelope to define better processes, strategies, and techniques for doing all of the above. Now, for the bad news: **Every process works in theory until you add people.**

And therein lies the heart of the challenge David set out to address with this book. Whenever we've switched from an old way of working to a new one, the journey has always required challenging the status quo and has been riddled with many little traps that can derail even the best-intentioned teams. This time is no different.

While many books tell you what to do, David takes a refreshing approach by telling you what *not* to do—informed by his 15-year journey as a product manager at start-ups and giant corporations. He starts by identifying the most common product traps in the first part of his book, then shows you practical steps for avoiding them—all illustrated with numerous real-world examples.

Untrapping Product Teams is a guidebook for anyone charged with building the next generation of digital products that matter.

We are living in a new era of product management, and David is one of the leading voices showing us the way. I hope you'll implement his practical guidance and help pave the way to a world with better products for all.

Welcome to the new era of product management.

—Ash Maurya, author, *Running Lean* and *Scaling Lean*
December 23, 2023
Austin, TX

Foreword by Jim Highsmith

The internet and graphical user interfaces (GUIs) transformed the software product landscape in the mid-1990s as IT applications transitioned from automating internal business processes to engaging external customers—a change that was obvious to everyone but whose implications were not always appreciated. Users then interacted with character-based, monochrome terminals; devices and computers were connected via wire; and internal operations software ran on a mainframe. The switch to customer-facing applications and technologies created a large demand for extra, specially educated product people and teams.

The game changed again in 2007 with the introduction of cloud computing, big data, iPhones, and social media. In the years 2022–2023, we were flooded with news of increased AI capabilities and the possibility of genuine quantum computing. So, interface design has evolved over the course of more than 30 years, from “green screens” to UI, UX, conversational AI, and driverless cars. At each juncture we needed more—a lot more—product people who understood the software development process and could integrate technological advances into effective product designs.

The fundamental competencies required to thrive in product development both now and in the uncertain times ahead are outlined in David Pereira’s book *Untrapping Product Teams*. Whereas others approach complexity with complex solutions, this book’s subtitle, “Simplify the Complexity of Creating Digital Products,” captures David’s basic idea. This is exemplified by his backlog management recommendation, which suggests most backlog items should be eliminated. This type of simplification shows up repeatedly in his book.

My all-time favorite quote comes from Dee Hock, former CEO of VISA:

“Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple, stupid behavior.”

David rails against what he refers to as “bullshit management,” which arises from complex rules and regulations that are often the product of a “fixed mindset.” If you want a prescription to follow, find another product management book. But if you want clarity of purpose and excellent practices that you can evolve supported by reality-based feedback, stick with David. He makes things “simple”—powerful and easy to understand—without descending into “simplistic” in a way that misleads.

One of the current issues that David addresses is the proliferation of “feature factories,” often fueled by the terribly misused “velocity” metric. In their drive to pump out features, delivery teams often put on blinders that shield them from anything beyond stripping the next stories off the backlog and hustling them through to completion. DevOps has exacerbated this trend. It’s like driving a car faster and faster down a freeway: “We’ve increased our speed (velocity)

three-fold in the last 50 miles; too bad we don't yet know our destination." The antidote to a feature factory is to concentrate on customer value, as David repeatedly reminds us. But is this really a problem? One research study indicated 80% of software features were rarely or never used, wasting nearly \$30 billion, according to a *Forbes* online article in 2019.

David's approach to generating customer value is multidimensional, from creating a value-oriented roadmap rather than one based on epics/features, to collaborating on defining release/sprint goals, to serious backlog pruning, to proposing a set of value-oriented rather than productivity-oriented metrics.

There are two fundamental approaches to product development: the prescriptive "plan-do" versus the adaptive "envision-explore." Subscribers to the fixed mindset think they can plan away uncertainty, whereas those who subscribe to a growth mindset understand uncertainty only yields to action through an exploratory process. David repeatedly charges in with ideas advocating a growth, or adaptive, mindset, particularly when describing the product discovery process. He starts off by saying, "It's tough to accept our ideas are mainly flawed," and then adds "I concluded what product discovery is: It's about the journey and not the plan." Fixed mindsets have a difficult time with flawed ideas and bumpy journeys.

Untrapping Product Teams contains a wealth of practical, usable ideas, in the form of narratives, checklists, outstanding questions, and solid principles. In addition, David's use of stories, experiences, and a conversational, first-person writing style engages the reader. While the primary audience for this book will be anyone with the word "product" in their job title, I think a much wider range of readers—from executives to software engineers—will benefit from David's ideas on product management.

—Jim Highsmith, author, storyteller,
and co-author of the Agile Manifesto
Lafayette, CO, 2023

Preface

This is the place where I openly share how *Untrapping Product Teams* came to exist. It gets personal. I give full credit to the many people who inspired me and helped bring this book to life.

Life has been generous to me. I come from a middle-class family in a small village in Brazil. My parents didn't finish high school but gave me the best education I could wish for. I got where I got not because I'm good, but because many people saw potential in me that I couldn't see myself.

I got my first job when I was 17 because Romero Silva, a computer store owner, believed I could learn to assemble computers. When I turned 24, I moved to São Paulo because Leandro Silveira, an experienced executive, perceived my communication skills as sharp enough to transform me into a product manager. Carlos Locoselli, a professor and mentor, invited me to give MBA lessons when I was 29 because he believed I had stories worth sharing. I moved to Germany because Emmanuel Wintzer (product lead) and Raimund Rix (chief product officer) bet on my potential.

How I came to write is another episode of unexpected opportunities.

Julio de Lima is a highly regarded influencer in the field of software quality assurance for me and many others. He inspired me to write my first post: "7 Key Points to Succeed as a Product Owner."¹ It took me hours, but I finally put the post out. Yet, I would not publish anything else for the next two years.

Anastasia, my wife, convinced me to try writing again. She told me on a walk in Gorky Park in Moscow, "You have a burning passion. You need to do something about it." That touched me, and I committed to writing one blog a month. Three years later, I had written more than 300 posts.

Product management is my passion, and I am especially inspired by three people: Marty Cagan, Melissa Perri, and Ash Maurya. After reading Marty Cagan's masterpiece, *Inspired*, my views on product management were transformed. Melissa Perri's book *Escaping the Build Trap* deeply spoke to me. Ash Maurya greatly impacted how I tackle ideas, prioritizing learning over delivery.

Why do I care so much about writing when you can find great material out there?

My epiphany came during the walk at Nymphenburg Park in Munich with Emmanuel, my first team lead in Germany. We talked about my recent blog, and I shared how much I enjoyed it, but I lacked ideas for new posts, which worried me.

1. You can read the original version in case you're curious. I've never edited this article to remind me how I started writing. It can be found at <https://medium.com/@davidavpereira/key-points-to-succeed-as-a-product-owner-e7643e3edf9e>.

Emmanuel untied the knot: “Have you considered writing about the don’ts? Most people write about what you should do, but only a few share what you shouldn’t.”

That was it! I didn’t want people to face the blows I had to take. I wanted to share what I did wrong and what I learned.

As my blog grew, I met many people who helped me become a better writer. A key person was and still is Maarten Dalmijn. He took time to review my rough drafts, challenged me, made suggestions, and helped me evolve. I owe him much for his genuine intention of helping.

Every person on stage has a strong team behind the curtains. I have someone who has always been by my side—Kevin, my brother. He is probably the only person on Earth who reads all my blog posts. He reviewed this book multiple times and was courageous enough to speak his mind and help me make the content sharper. I cannot find words to express my gratitude to him.

I danced around the book idea for a long time but didn’t feel ready to commit. April 20, 2023, was the day that changed. I gave a keynote speech, “Untrapping Product Teams,” at the Product Tank Munich. After my address, Ioana Ognibeni, one of the best product managers I know, insisted I had to write a book. She said that I had something to say that people needed to know. Her words got to me. Three days later, I started writing the book.

Ioana and I worked together for a year. Her passion for product management amazed me. She said, “The world needs a book talking about reality, and you know how to talk about it in a way that makes people listen.” This sentence inspired me to write the book you’re reading now.

After talking to Ioana, I wanted to share what I had learned throughout my career. I often learned lessons the hard way. I faced traps and didn’t know how to overcome them, or even worse, I was unaware I was trapped. I thought more people could benefit from my experience. However, writing a book is no easy task. It goes way beyond blogging.

I was lucky to find amazing people to help me sharpen the book. Special thanks go to Maria Chec, who encouraged me to keep my authenticity. Simonetta Batteiger, who read the manuscript on a flight from Germany to Boston, helped me identify confusing content and simplify the message.

I’m tremendously grateful for Ash Maurya, who wrote a generous foreword and took the time to encourage me to keep my original title and shared the behind-the-scenes stories of his own book, *Running Lean*. Beyond that, his approach to treating learning as an unfair advantage drove me to craft this book differently, learning from the beta readers since the early book stages.

I must give a massive special thanks to Jim Highsmith, co-author of the Agile Manifesto, who invested significant time reviewing the content, helping me improve it, and doing me the honor of writing a wonderful foreword for this book.

Mike Cohn encouraged me to push this book forward, which boosted my energy to go the extra mile. Having his support supercharged my inspiration.

Special thanks to the beta readers who spent their time reading early versions of the book. I owe you a lot: Sandra Hinz, Stefan Wolpers, Alexej Antropov, Jennifer Forbes, Si Chen, Nuno Santos, Atanas Kostadinov, Corinna Hammersting, Sebastian Borggrewe, Emiliano Saad, Irene Liakos, Felipe Borgonovi, Mike Berman, and Vira Chesnokova.

Of course, I could not forget the outstanding editorial team behind the production of this book. I wholeheartedly thank Haze Humbert for her strong support from end to end, Menka Mehta for all the assistance provided, Julie Nahil and Jill Hobbs for improving the book's readability and flow, and Jayaprakash P. for coordinating the book project.

This book wouldn't exist without all these people. I hope *Untrapping Product Teams* empowers you to rock the world!

Register your copy of *Untrapping Product Teams* on the InformIT site for convenient access to updates and/or corrections as they become available. To start the registration process, go to informit.com/register and log in or create an account. Enter the product ISBN (9780135335383) and click Submit. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

This page intentionally left blank

About the Author



David Pereira is a product leader with more than 15 years of experience. He's sharpened his skills by leading diverse teams, from start-ups to giant corporations. Since 2020, he has openly shared his mistakes, failures, and insights on product management, reaching more than 10 million readers worldwide. His thought-provoking courses had 15,000-plus satisfied students across more than 120 countries.

The numbers don't drive David, but the outcomes do. He often receives messages from people worldwide sharing stories on how his content helped them overcome a challenge, get a new job, receive a promotion, or even start a business. Such messages touch and inspire him.

Let his unique expertise inspire your journey.

Feel free to drop a message at contact@d-pereira.com.

This page intentionally left blank

1

How Common Ways of Working Trap or Untrap Teams

Expectations and reality tend to diverge in the realm of digital product management. Let us imagine a few scenarios to compare expectations with reality.

Suppose you become a product manager and are excited to create products customers love and help your business thrive. With such a motivation, nothing can stop you. But suddenly, you bump into the corporate firewall. When you face prescriptive roadmaps, extensive timelines, and output commitment, you get sucked into backlog management, quickly getting demoted to a backlog manager.

This kind of disappointment isn't exclusive to product managers. Imagine you're a software engineer. You're passionate about crafting clean code and building scalable solutions. You want to bring your best to the team, but then you realize that shipping features at the speed of light matters most. Pressured, you cut quality to meet deadlines. Now, you're no longer a software engineer. Without realizing it, you've become a coder. Your code stinks and your solutions aren't scalable. Despite initially delivering on time and budget, you're surprised because nobody uses the shiny features you gave your heart and soul to create.

What would happen if you were a product designer? You'd also get a gold pass to the club of frustrated people. You aim to create intuitive designs and outstanding user experiences, but your motivation lasts only until reality hits you. Astounded, you realize that success means creating pixel-perfect prototypes that business stakeholders love and approve, even though they don't represent your customers.

Do any of these situations ring a bell to you? If not, we may live in different worlds, because I've been part of them from South America to Europe, from start-ups to well-established corporations. The question is, what do you do when you face such situations?

When people lack skills, some may complain and do nothing about it. This attitude is what I call becoming the victim of the circumstances.¹ In contrast, with the proper skill set, you can challenge the status quo and transform your situation by engaging in a step-by-step journey.

Simple actions can help you overcome the common traps described earlier. To equip you with the skills to make that possible, I invite you to join me in a conversation about how teams typically work, because that reality has a tremendous impact on what they can achieve.

Let's explore the dynamics frequently present in feature factory companies.

How Feature Factory Companies Work

Not every company is the same. Every organization has its particularities and dynamics, though they all share some common traits. Understanding your company's unique dynamics enables you to identify opportunities to simplify how its product teams work.

First, let's look at companies' typical structure. Companies have multiple departments—for example, management, marketing, operations, customer service, product, and so on. Sometimes, these departments are called teams or areas. What separates departments from one another are their responsibilities and way of working. Departments often work inside their bubble and coordinate tasks with other departments when necessary.

In larger companies, the complexity increases with divisions and units under different leaderships. The relationship between such divisions may be quite distant, and collaboration becomes daunting. For our analysis, we'll stick with a small company. Figure 1.1 depicts some typical departments.

What all departments have in common are ideas, and each department often perceives its ideas as important and urgent. Some ideas relate to their work, others to the company's core product or service.

The more unclear the company's vision and strategy are, the harder it becomes to decide what to do with endless ideas.

1. Scientifically, being a "victim of the circumstances" relates to the person's locus of control. Having an internal locus of control means that the person thinks they control their own life and the outcomes of events, while having an external locus of control means embracing the opposite view. Julian B. Rotter developed this concept in 1954, and it has since become a key aspect of personality psychology.



Figure 1.1

Example of company departments

This situation is overwhelming because ideas have different objectives and, in most cases, have no connection with each other. In turn, the excess of ideas and pressure for delivery challenge product teams. Meeting expectations requires interactions with all departments, understanding their wishes, and doing something about them.

Collaboration between Product Teams and Business Stakeholders

Creating a backlog is a common way of dealing with the ideas suggested by the business stakeholders (i.e., the business members interested in your product or service). This method represents a dangerous trap for product teams. Collecting ideas and adding them mindlessly to a backlog is easy—but having a gigantic list of unrelated ideas creates unbearable expectations.

The depth of ideas varies significantly. Sometimes, they are on a milestone level, such as “Expand our product to a new audience.” At other times, ideas are related to specific wishes, such as “Allow customers to export their orders of last 90 days.” This diversity complicates prioritization. Yet, the typical approach is to put everything in the same bucket and have long discussions until everyone agrees on what comes first.

With extensive backlogs, more business stakeholders than you can handle will knock on your door, asking when you plan to deliver on their ideas. That forces you to coordinate tasks with everyone, complicating collaboration as you juggle multiple topics.

Figure 1.2 illustrates the complex dynamics between product teams, departments, and the backlog. The bigger the company, the more complex this situation becomes.

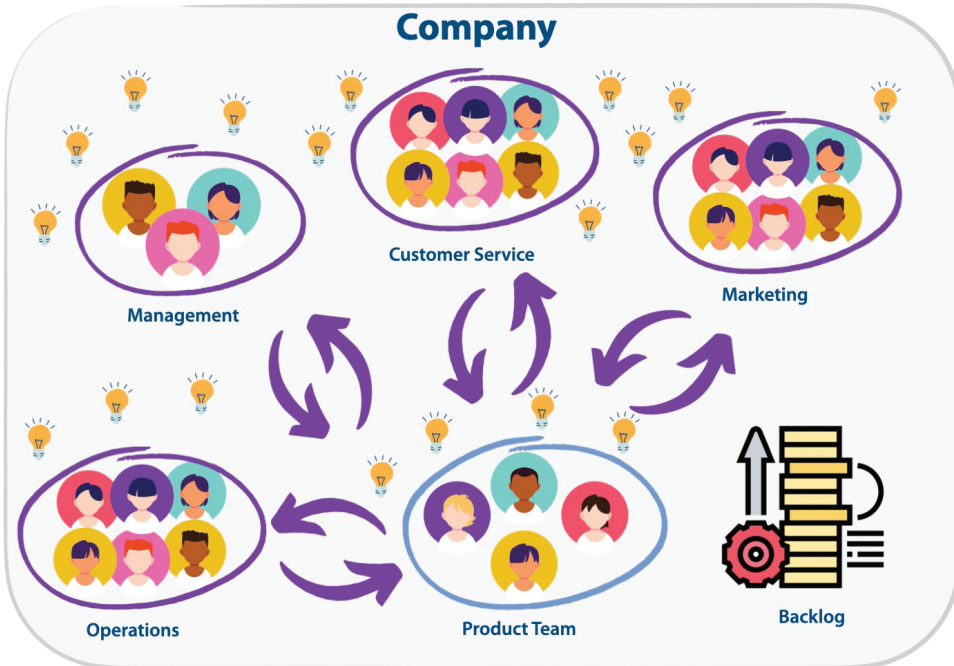


Figure 1.2

Product teams' interactions with business stakeholders and the backlog

When you look at Figure 1.2, what's wrong with it? Take a minute to reflect on it.

Products exist to serve customers, not business stakeholders. When the people creating the product don't interact with customers, creating useless products is the potential result.

It shocks me how often companies develop solutions for their customers without interacting with them. It's easy for us to fall in love with ideas, become blind to reality, or, even worse, forget about it. The results are often undesired.

Releasing New Features Nobody Cares About

How's your experience releasing new features? Maybe it's a hit or miss. Let's continue our conversation about the sad reality of feature factory teams.

After months of hard work and exhaustive coordination, the product team got a new feature out of the backlog. Everyone from the team and business loves it. The new shiny feature is ready for customers, but something unexpected happens. Customers who interact with the new feature don't understand its purpose and cannot benefit from it. Confused, customers reject the new shiny feature beloved by business stakeholders, and inevitably everyone becomes frustrated (Figure 1.3).

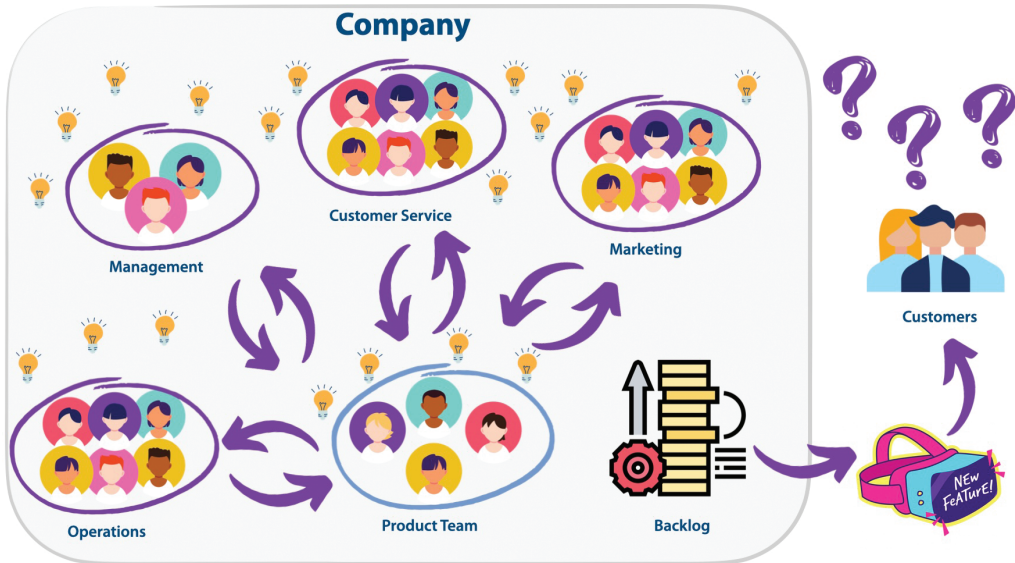


Figure 1.3
Building features customers don't need

Creating solutions companies love, but customers couldn't care less about, isn't why product teams exist. Yet, that happens more often than it should. I call it a bug, not a feature. As with all critical bugs, it requires a hotfix.

If the feature factory is a bug, what would fix that? You cannot expect a simple fix, but fostering empowered teams with collaborative flows will transform the situation. Marty Cagan, SVPG Partner, wrote a blog post defining empowered teams²:

Great teams are comprised of ordinary people that are empowered and inspired. They are empowered to solve hard problems in ways their customers love, yet work for their business. They are inspired with ideas and techniques for quickly evaluating those ideas to discover solutions that work: they are valuable, usable, feasible and viable.

2. Marty Cagan, "Empowered Teams," www.svpg.com/empowered-product-teams/.

Empowered teams have significantly higher chances than feature factory teams to create value. Yet, enabling empowered teams isn't trivial. We will discuss what it takes to get there in Chapter 10. For now, let's focus on the collaboration challenges.

Over the years, I've noticed two standard product development flows spread across companies regardless of their framework:

1. **Coordinative** → Team members spend significant time coordinating activities among themselves, stakeholders, and other teams. Most of their energy goes into organizing how to get the work done. This approach aims to avoid mistakes and failures, which forces teams to be rigid with their development flow. It becomes a "strict" contract because someone gets the blame when something goes wrong. Plans become the ultimate goals because nobody knows what they are fighting for.
2. **Collaborative** → Team members focus on collaborating to use their current knowledge to uncover promising opportunities. The ultimate goal is to create value for customers and the business. The team is flexible with how they get the job done while focusing on driving value. Trust is the basis for the collaborative approach. When something derails, the team takes responsibility and jointly finds a solution. Teams start with simplified plans, progress toward the *unknown*, and adapt the following steps based on the evidence. An overarching goal guides their decisions, ultimately creating value for everyone involved.

The product development flow dramatically contributes to teams becoming empowered or trapped in feature factories. Let's explore each of these two possible flows in detail.

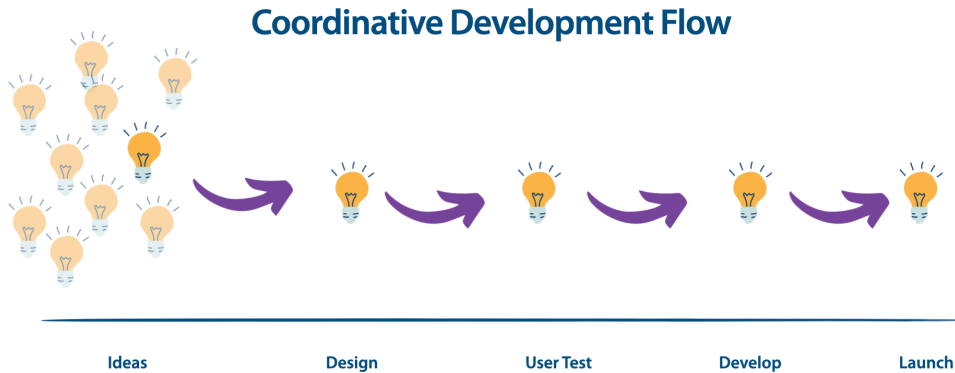
Coordinative Flow: A Logical Way of Working with Unexpected Results

How do you transform an idea into something valuable?

That's one of the most important questions for companies. A poor answer leads to waste and demotivation.

The beginning of most product development flows is almost always the same. You probably have several ideas and a limited capacity to transform them into reality. The question is, how do you progress?

A coordinative flow aims to carry out ideas from beginning to end, treating each step as a strictly different phase. Figure 1.4 depicts this approach.

**Figure 1.4**

How an idea moves from beginning to launch with a coordinative flow

The coordinative flow starts with prioritization, aiming to find the most *promising* idea. However, that's easier said than done because several discussion rounds will occur. When you say *yes* to an idea, you're saying *no* to numerous others, and almost no business stakeholder accepts that answer easily.

Prioritization is one of the reasons product managers struggle to sleep. Getting everyone committed to one objective is tough. In a coordinative flow, prioritization will take weeks, if not months.

Design

The design phase starts once you define what the team will work on. The result is often a high-fidelity prototype that business stakeholders approve. This approach is dangerous because software engineers and customers tend to be left out of it. Sadly, the solution becomes the focus, not the outcome.

You may assimilate the design phase as a traditional waterfall process. Yet, having extensive design phases happens with agile methodologies as well. Whether you work with Scrum, Kanban, Scrumban, or something else, it doesn't matter: The result can be the same. Often, product designers aren't part of the product team, and teams receive detailed solutions to implement instead of opportunities to explore.

Designing solutions outside the product team is another disturbing bug that requires a hotfix. We will address that later in this book (see Chapter 4).

User Test

After much coordination, business stakeholders finally approve the design, and it's time to test it with potential customers.

The results are probably compromised because everyone already loves the solution. Sadly, falling prey to confirmation bias isn't the exception, but rather the typical outcome. Given their passion for the solution, product designers search for positive signs—and unsurprisingly, they find them. They may accept minor solution tweaks, but no solution pivot or drop will happen in this phase.

Develop

After product designers *confirm* the high-fidelity prototype makes sense to end users, it's time to develop the solution. Product designers throw the specs over the fence and hope software engineers do the job right. Of course, software engineers aren't likely to welcome the solution with wide-open arms because they weren't part of the previous steps. Even so, it becomes their job to transform the high-fidelity prototype into a working solution.

During the development phase, product designers and software engineers often come in conflict. Software engineers challenge the design as they see potential alternatives, and product designers want the implementation to follow the high-fidelity prototype strictly. Together, they find compromises, and none of them are happy about it.

The sad part is the extreme focus on the output. Nobody talks about the customer at this stage. Yet, they keep working on it until the solution is ready for the big moment.

Launch

Given the amount of coordination needed, it takes months to transform an idea into a solution. You shouldn't be surprised when something takes half a year. Each phase is strictly defined and has many steps to ensure a perfect solution by the end of it. Yet, something unexpected happens when you launch the solution.

Despite all of the internal enthusiasm and love for the newest fancy solution, customers don't engage with it—and you have no idea why.

You're sure the design went through rigid feedback loops, the color tonalities are correct, and management approved. The user test took place with at least ten potential customers, and they *confirmed* they would use the solution because it looked nice to them. Moving from prototype to solution was challenging, but you did it. You're clueless about what went wrong.

You wanted to celebrate the feature launch. After all, getting to this point required many exhausting discussions and took a toll on your nerves. Inexplicably, you ended with the fanciest piece of crap ever created (Figure 1.5). Sorry about my vocabulary, but recognizing reality requires facing it as harshly as possible.

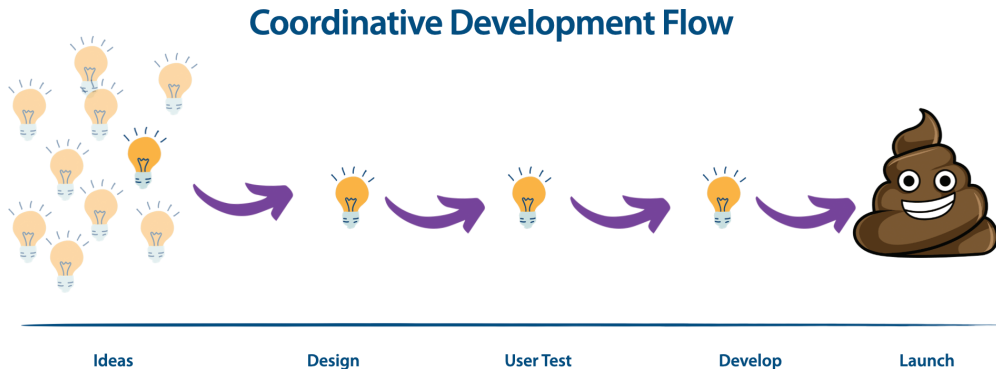


Figure 1.5

Launching something and getting disappointed with a coordinative flow

What shocks me isn't winding up in this tragic situation. I ended there more times than I can count, but I learned my lesson. The question is, what do you do after you face an undesired outcome?

The most common answer makes little sense to me. Go back to prioritization, pick another idea, and do it all over again. When you follow the same approach, chances are high that you will face the same results again.

The coordinative flow forces teams to focus on outputs over outcomes, diminishing them to feature factories.

Nine Out of Ten Ideas Will Fail

Our success rate is worse than we can imagine. Look at start-ups. Ninety percent of them don't last more than five years.³ Ideas suffer much the same fate. Curiously, that goes unnoticed as teams invest much time figuring out how to reduce development time. I see value in this matter, though I perceive another question as more pressing: *How fast can you drop bad ideas?*

We assume our ideas are good, but reality shows us otherwise. Yet, we insist on following the same approach repeatedly. No wonder we face undesired results.

3. "90% of Startups Fail: Here's What You Need to Know about the 10%," *Forbes*, www.forbes.com/sites/neilpatel/2015/01/16/90-of-startups-will-fail-heres-what-you-need-to-know-about-the-10/.

Sound product management requires adapting based on learning. It's fine to get things wrong. It's not fine to ignore reality.

Collaboration over coordination is the principle that can get you out of this trap. Instead of making your development flow rigid and complex, you will benefit by making it simple and flexible.

Let's explore a different way of working that increases the odds of driving value faster.

Collaborative Flow: A Simple Way of Working with Outstanding Results

Nobody deserves to waste time. I know how much it hurts to see your work leading nowhere. After many years on the road, I learned that failures are inevitable. Instead of adding steps to prevent failures, identifying and quickly dropping flawed ideas make more sense.

Everything I share in this book is simple because I'm not good at dealing with complexity. Day in and day out, I strive to remove complexity by adding simplicity. That's how I uncovered ways of working that increase the chances of creating value sooner.

The more you simplify, the sooner you can move from a feature factory to an empowered team.

Unlike the coordinative flow, the collaborative flow focuses on iterations instead of phases. Each iteration enhances learning, empowering teams to choose between investing further or dropping the idea.

Figure 1.5 illustrates how teams start with several ideas and collaborate to identify what drives value while dropping what doesn't.

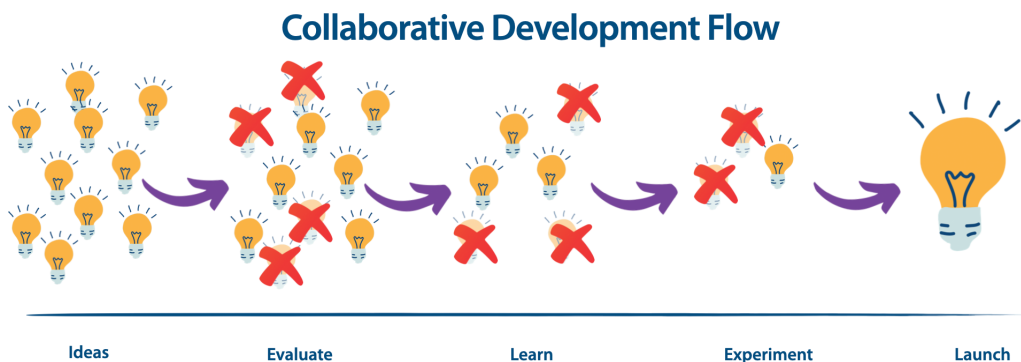


Figure 1.6

Dropping bad ideas fast enough and focusing on promising ones with a collaborative flow

Let's understand the different iterations of a collaborative flow and how that leads to empowerment.

Evaluate

The beginning of a collaborative flow is the same as for a coordinative flow. You've got plenty of ideas, and everyone wants everything done by yesterday. The trick isn't to identify the most *promising* ideas upfront, but rather to evaluate all of them and drop the misfitting ones. *Dropping ideas gives you freedom because you've got fewer expectations to manage.*

To drop ideas, confront them with your strategy. Here are the questions you should ask:

- How does it get us closer to our product vision?
- How does it relate to our product strategy?
- How does it contribute to our current objectives?
- How does it deliver on our value proposition?

Drop the idea if you lack answers to any of these questions. You may miss some of the attributes I mentioned (vision, strategy, objectives, value proposition) and struggle to assess your idea against them. You've got to do your homework (read Chapter 5 to set your product strategy).

Don't invest time in ideas unrelated to your strategy. By that, I mean don't increase the size of your backlog, but instead make your trash bin bigger.

You'll be tempted to park your idea somewhere and eventually return to it. Don't do that, because it will distract you. Whatever is relevant to your customers and business will return to you.

Idea evaluation should take a couple of hours, not more than that. Collaboration is essential, so get key business stakeholders to do this exercise with the product team. It's not about defending ideas, but rather about checking how they fit your strategy.

Strive to identify fitting ideas and drop unrelated ones pragmatically.

Learn

The learning iteration starts with ideas fitting your strategy, but that doesn't mean jumping straight to implementation. You should drop ideas your customers don't desire, the business cannot support, you don't have the technology to develop, or it's unethical to pursue. Keep it simple, and ask the following questions about each remaining idea:

- How much do customers want it? (Desirability)
- How does the business benefit? (Viability)

- How well can we deliver it? (Feasibility)
- How right is doing it? (Ethics)

Figure 1.7 represents an adapted version of the Innovation Trinity popularized by IDEO,⁴ also known as a product Venn diagram. The overlapping areas highlight the sweet spot of promising ideas based on the preceding questions.

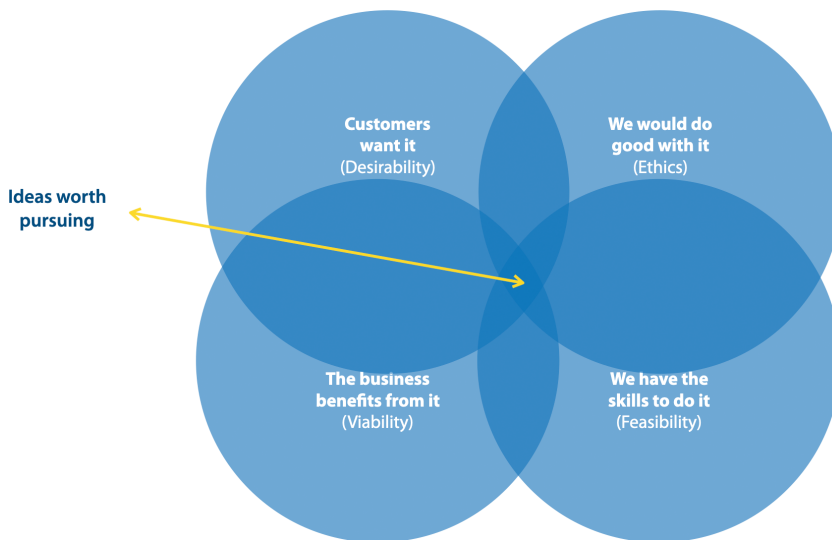


Figure 1.7

An adaption of the Innovation Trinity showing the sweet spot of promising ideas

The sooner you drop bad ideas, the sooner you can focus on the promising ones. Answering critical questions will help you with that. I like how Marty Cagan (2018) approaches this issue. He uses the following questions to address critical risks:

- Will the customers buy this or choose to use it? (Value risk)
- Can the user figure out how to use it? (Usability risk)
- Can we build it? (Feasibility risk)
- Does this solution work for our business? (Business viability risk)

You may not find answers to each of these questions, but you will have assumptions. Invest some time testing the critical assumptions (business critical and weak evidence), and decide which ideas are worth pursuing.

4. IDEO popularized the Innovation Trinity, also known as a product Venn diagram. I adapted this version to add ethics, as I see the importance of taking this factor into account. See <https://blog.leanstack.com/lean-startup-or-business-model-design-or-design-thinking-is-the-wrong-question/>.

This iteration should take a few days, but not more than that. Use qualitative information to determine how to progress. You can do interviews with customers and business stakeholders, low-fidelity prototype testing, and quick technical experiments as needed.

The goal is to find evidence confirming ideas are desirable, feasible, and viable (refer to Chapter 7 for more information on how to do this). If one of these characteristics is missing, drop the idea and move on.

A common trap is focusing on viability and feasibility. Just because you can do something, that doesn't mean you should do it. Investing in an idea without evidence of desirability is no more than a bet. The result is often a feature nobody uses. Proceed only with the ideas that have evidence strong enough to justify the investment you're about to make.

It's fundamental to collaborate with business stakeholders and customers. Otherwise, you will end up with undesired results.

You'll drop 30% to 50% of the remaining ideas.⁵ If your numbers are out of this range, either you're too strict or you're too loose. I'd recommend reviewing your evidence and decisions.

Experiment

After learning about the key aspects of your ideas, it's time to run more robust experiments. You want to test which solutions can deliver the potential results. Exploring a few alternatives and sticking with the most promising ones is essential.

It's all too common to pick one solution and go all in with it. I discourage you from following this path, because it quickly leads to commitment escalation. As humans, the more invested we are in something, the more willingly we invest in it.

You need to choose which experiment makes more sense to you. Your ultimate objective is to have solid information that justifies investing further. For that, you will need a combination of qualitative and quantitative data.

It's not within the scope of this book to cover the diverse product experiments you can apply. I recommend reading *Testing Business Ideas* by David Bland to learn more about this topic.

Here's my secret: I like using tech debt (software quality compromises requiring future rework) as a tool to accelerate learning. The goal is to hack a quick and dirty solution, get it live to a small portion of your audience, and gather evidence on how that helps them get the job done. I must warn you, though, software engineers are often opposed to increasing tech debt. They tend to be strict about creating this kind of tech debt, because many product managers do not support them in paying it off on time.

5. It's hard to accept that. But over time, I learned that many ideas cannot resist an encounter with reality. When I tested ideas with an open mind, I ended up dropping almost half of my ideas. Yet, when I neglected evidence, I ended up creating something that nobody needed.

Prudent use of tech debt is a tool⁶ (more on that in Chapter 8). It's like getting a mortgage. You go to the bank, get a mortgage, and acquire your dream house. You collected the desired value fast, but then you owe the bank and should pay the debt off. If you accumulate more debt, you may go bankrupt. The same thing happens with tech debt. It enables you to reduce the time to collect value, but whenever it's proven worthwhile, you must pay the debt off before getting a new loan.

You will drop another 30% to 50% of the remaining ideas during the experiment iteration. The reasons will vary, but it turns out that customers often show unexpected behaviors, which enables you to review how worthy your idea is. Not all ideas justify the investment despite being desirable, feasible, and viable at first glance.

The critical part of experimentation is to drop solutions that don't work. Feature factory teams can't do that because delivering the output is their goal. But empowered teams can because achieving the outcome is their success metric.

Whenever you want to drop an idea, involve your business stakeholders to reach this conclusion collaboratively. Evidence will support your decision, and involving business stakeholders will get their buy-in (read the discussion of "opinions over evidence" in Chapter 2).

Launch

Ideas that survive the experiment iteration are the prominent ones. In the previous iteration, you built to learn. Now, you build to scale. Paying the tech debt off before you make the solution available to your whole audience or jump to your next opportunity is fundamental.

Unlike with a coordinative flow, applying a collaborative flow will help you drop bad ideas faster. This method benefits from the power of collaboration over coordination.

Be aware that not all companies are comfortable with a highly collaborative flow. For example, in Germany, many companies long for yearly plans with feature roadmaps and prescriptive timelines. Sadly, those plans create an exhaustive need for coordination to get everything progressing. Yet, you can foster change gradually. When facing such situations, you could take a step-by-step journey to help companies uncover better ways of creating value (read Chapter 4 to get insights on how to act).

I have one thought about teams that are able to create value faster than others, and it's a rather obvious one: You have to do what most teams don't do to achieve the results most teams don't get. Move away from coordinative flows, and do your best to focus on collaborative flows instead.

The more coordination you have to endure, the less time you have to focus on uncovering what creates value. Don't force your teams to behave as feature factories.

6. Martin Fowler categorizes technical debt into four quadrants. The one I'm recommending relates to deliberate and prudent use. See <https://martinfowler.com/bliki/TechnicalDebtQuadrant.html>.

The more effectively your teams can collaborate, the faster they can adapt, learn, drop bad ideas, and focus on promising ones. That's the power of enabling empowered teams.

Key Takeaways

- The first step to untrapping your team is reviewing your current situation. Understanding the dynamics can reveal opportunities to simplify your development flow.
- Symptoms of a feature factory include a lack of goals, output obsession, delivering solutions that solve no problem, unclear direction, unwillingness to drop ideas, and inattention to results. The opposite of a feature factory is an empowered team that focuses on outcomes and is eager to inspect and adapt continuously.
- The closer you are to a coordinative development flow, the longer it takes to deliver value and the more waste you create. Coordinative flows unwittingly transform plans into goals.
- The closer you are to a collaborative development flow, the sooner you create value and the less waste you produce. Collaboration helps you adapt plans to reach your goals when reality makes your plan obsolete.
- When you fully understand the product development flow, you can foster changes step by step. The first step is collaborating with business stakeholders and team members to recognize what's unnecessarily complex. Equipped with that knowledge, you can gain support and collaborate to simplify your work.

In this chapter, we discussed how teams commonly work. That's a critical piece of the puzzle, but it isn't the only reason teams struggle to deliver value. Chapter 2 covers another fundamental aspect: how the mindset affects digital product creation.

This page intentionally left blank

Index

Numbers

1:1 interactions, random, 180

A

A even over B, product principles, 188
A instead of B, product principles, 188
A over B, product principles, 188
accelerated learning, tech debt, 19–20
accountability

- company dynamics, 199
- leadership, 199
- ownership and, 206–207
- product strategies, 211
- teams, 161–162
- value creation, 44–45

acquisitions, CAC, 150
actionable metrics, 139

- activation metrics, 150
- activation rates, 150
- AI insights, 149
- alerts, 149
- average session duration, 149
- bounce rates, 149
- CAC, 150
- CES, 142, 144
- churn rates, 150
- CLV, 150
- conversion rates, 149
- CTR, 149
- customer retention rates, 149
- data traps, 140–141
- defining, 144–146
- engagement metrics, 149

feature adoptions, 150
gross margins, 150
identifying, 142
input/output metrics, 143–144
laggard metrics, 141–143, 151
leading metrics, 141–143, 151
measuring metric results, 146–147
monetary metrics, 150
MRR, 150
onboarding, 145, 150
referral rates, 150
return visits, 149
scheduled reports, 148
sharing metric results, 147–148
sign-up rates, 150
time saving techniques, 148–149
unique visitors, 149
vanity metrics, 140–141

activation metrics, 150
activation rates, 150
adopting features, 49, 150
adoption life cycles, 67–68
Adrià, Ferran, 56–61, 230
agile approaches (empowered teams), dual-track, 158–159
Agile Coaches, 168–169
agile rulers, 168
Agile Swarming Conference (2023), reality check surveys, 26–27
agile teams, velocity, 42
AI insights, actionable metrics, 149
alerts, actionable metrics, 149
alignment

- collaboration, 177–179
- cycles, 127–129

- goal-based alignment, 127–128
 - product delivery, 127–129, 134–135
 - product principles, 194
 - analysis paralysis, 211
 - analysis (product strategies), progress over, 208
 - Apple
 - iPhones, product vision, 66
 - iTunes, 226
 - approving output, value creation, 44–45
 - assessing
 - company dynamics, 197–204
 - product principles, 189
 - assumptions
 - confirming, 114
 - defined, 111
 - discovery journeys, 110–114, 118
 - hypotheses, 111
 - identifying, 111–113, 216–217
 - ignoring, 114
 - leap of faith assumptions, 114
 - opportunity solution trees, 110
 - prioritizing, 113–114
 - product discovery, 216–218
 - product experiments, 114–116
 - testing, 118, 218
 - attractive but inviable, 160
 - audiences (target), identifying, 66–68
 - authenticity, building relationships, 180
 - author's resources
 - blog, 3
 - feedback, sending, 231
 - Lean Canvas, 74
 - product adoption life cycles, 68
 - roadmap templates, 83
 - stakeholder matrix, 173
 - Substack, 3
 - average session duration, 149
 - avoiding failures, value creation, 50
- B**
- B2B scenarios, product discovery, 216
 - B2C scenarios, product discovery, 215
 - backlogs
 - cleaning, 124–125
 - Definitions of Ready, 123–124
 - due dates, 124
 - item formats, 123–124
 - lean product backlogs, 123
 - managing, 7, 121–122
 - new products, 122–123
 - product delivery, 121–125, 137, 226–227
 - product managers versus backlog managers, 162–164
 - product teams, 10
 - running products, 123
 - value, determining, 121–122
 - value creation, 40–41
 - bad product vision, 70–71
 - balance
 - importance of, 50
 - product discovery, 60
 - bandwagon fallacy, 94–95
 - Bassino, Nacho, 65
 - Bastow, Janna, 82–85
 - biased product discovery, 91, 102
 - bandwagon fallacy, 94–95
 - commitment escalation, 92–93, 118
 - confirmation bias, 92, 118
 - sunk-cost fallacy, 93–94
 - Bishop, Bill, 75
 - Bland, David, 19
 - assumption matrix, 113
 - Testing Business Ideas*, 19
 - blogs
 - author's blog, 3
 - status-updating micro-blogging, 226
 - Blue Ocean strategy, 80–82
 - bottlenecks, value creation, 47–48
 - bounce rates, 149
 - bridge communication, bottlenecks, value creation, 47–48
 - Bryar, Collin, 143
 - building relationships, 171
 - alignment, 177–179
 - authenticity, 180
 - consistency, 175
 - face-to-face engagement, 180–181
 - in-person exchanges, 180
 - random 1:1 interactions, 180
 - shared understanding, 175–177
 - sharing progress, 176–177
 - spontaneity, 180
 - stakeholders, 172–181
 - structured meetings, 177
 - transparency, 176
 - building to learn, then to scale (product principles), 194
 - “bullshit management”
 - checklist, 51–52
 - defined, 38–39
 - Bullshit Management*, 38
 - Burbn app, 225
 - Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, 74
 - business models
 - freemium, 73
 - marketplace-based, 74
 - on-demand, 73
 - SaaS, 73
 - subscription-based, 73

- C**
- CAB (Customer Advisory Boards), 216
 - CAC (Customer Acquisition Costs), 150
 - Cagan, Marty
 - crafting product vision, 72
 - critical risk assessments, 18
 - empowered teams, 11–12
 - Canvas, Lean, 66, 72–73
 - channels, 76
 - cost structures, 77
 - crafting, 74–79
 - customer segments, 75
 - key metrics, 77
 - Netflix, 79
 - problems, defining, 75–76
 - revenue streams, 77
 - Slack, 79
 - solutions, 76
 - Substack, 75–79
 - unfair advantages, 78–79
 - unique value propositions, 76
 - car dealers
 - assumptions, identifying, 111–113
 - customer journeys, 102–104
 - discovery journeys, 102–104
 - evidence, 116–117
 - opportunity solution trees, 108–109
 - product discovery, 102
 - user story mapping, 106–107
 - Carr, Bill, 143
 - Catmull, Ed, 53, 153
 - CEO (Chief Executive Officers)
 - product leadership, 201
 - product strategies, 209–210
 - CES (Customer Effect Scores), 142, 144
 - channels, Lean Canvas, 76
 - Chief Executive Officers (CEO)
 - product leadership, 201
 - product strategies, 209–210
 - Chief Operations Officers (COO), product principles, 189
 - Chief Product and Technology Officers (CPTO), product leadership, 201
 - Chief Product Officers (CPO), product leadership, 201
 - Chief Technology Officers (CTO), product leadership, 201–202
 - churn rates, 150
 - circumstances, victim of the, 8
 - clarifying goals, discovery journeys, 101–102
 - clarity
 - customer journeys, 104
 - of goals, product strategies, 208–210
 - of importance, product discovery, 214–215
 - cleaning backlogs, 124–125
 - Click-Through Rates (CTR), 149
 - clusters, stakeholder matrix, 174
 - clutter (product delivery), avoiding, 223–224
 - CLV (Customer Lifetime Value), 150
 - coders, 165–167
 - collaboration, 171
 - alignment, 177–179
 - authenticity, 180
 - consistency, 175
 - face-to-face engagement, 180–181
 - in-person exchanges, 180
 - problem solving, 195
 - product leadership, 201–202
 - product principles, 192
 - product strategies, 210–211
 - random 1:1 interactions, 180
 - shared understanding, 175–177
 - sharing progress, 176–177
 - spontaneity, 180
 - stakeholders, 172–181
 - structured meetings, 177
 - teams, 154–155, 159
 - transparency, 176
 - collaborative development flows, 12, 16
 - evaluation phase, 17–18
 - experimentation phase, 19–20
 - Innovation Trinity, 18
 - launch phase, 20–21
 - learning iteration, 17–19
 - product Venn diagrams, 18
 - tech debt, 19–20
 - collaborative teams, 154–155, 159
 - commitment escalation, 92, 118
 - common team setups, 154–155
 - communication
 - authenticity, 180
 - consistency, 175
 - face-to-face engagement, 180–181
 - in-person exchanges, 180
 - random 1:1 interactions, 180
 - sharing progress, 176–177
 - spontaneity, 180
 - stakeholders, 172–181
 - structured meetings, 177
 - teams, 165
 - transparency, 176
 - company departments, feature factories, 8–9
 - company dynamics, 197
 - accountability, 199
 - health checks, 198
 - leadership, 198–199
 - product leadership, 201–202
 - product perception, 200
 - risk tolerance, 203–204
 - team topologies, 202
 - compelling visions, 69–70
 - consensus-driven decisions, value creation, 43–44
 - concierge-driven ideation technique, 110

- configuring teams
 - accountability, 161–162
 - collaborative teams, 154–155, 159
 - common setups, 154–155
 - coordinative teams, 154–155
 - dependent teams, 154–155, 156–157
 - empowered teams, 154–155, 158–159, 169
 - empowered-but teams, 154–155
 - feature factories, 154–155
 - independent teams, 154–155, 157–158
 - roles/responsibilities, 160–162
 - size of, 159
 - trapped teams, 154–155
 - confirmation
 - assumptions, 114
 - bias, 92, 118
 - consensus teams, 162
 - consistency, shared understanding, 175
 - context switching, product strategies, 209–210
 - Continuous Discovery Habits*, 103–104
 - conversion rates, 149
 - COO (Chief Operations Officers), product principles, 189
 - coordination
 - product leadership, 201–202
 - teams, 154–155
 - coordinative development flows, 12, 21
 - design phase, 13
 - development phase, 14
 - launch phase, 14–15
 - prioritization, 13
 - user tests, 14
 - coordinative teams, 154–155
 - cost structures, Lean Canvas, 77
 - CPO (Chief Product Officers), product leadership, 201
 - CPTO (Chief Product and Technology Officers), product leadership, 201
 - crafting
 - Lean Canvas, 74–79
 - product principles, 186–188
 - product vision, 71–72
 - crazy eights ideation technique, 109
 - creating value, 37
 - accountability, 44–45
 - balance, importance of, 50
 - bottlenecks, 47–48
 - bridge communication, 47–48
 - “bullshit management”, 38–39, 51–52
 - consensus-driven decisions, 43–44
 - drivers, 38
 - failure avoidance, 50
 - feature reports, 49
 - meeting marathons, 45–46
 - observing interviews, 49
 - opinion-driven decisions, 48–49
 - output approval, 44–45
 - output reports, 42
 - passengers, 38
 - product backlogs, 40–41
 - requirements gathering, 39–40
 - saying “no” strategies, 46–47
 - critical risk assessments, 18
 - Croll, Alistair, 149
 - Crossing the Chasm*, 68
 - CTO (Chief Technology Officers), product leadership, 201–202
 - CTR (Click-Through Rates), 149
 - Curves, Value, 66
 - Blue Ocean strategy, 80–82
 - creating, 80
 - example of, 81
 - plotting, 81–82
 - Customer Acquisition Costs (CAC), 150
 - Customer Advisory Boards (CAB), 216
 - Customer Effect Scores (CES), 142, 144
 - Customer Lifetime Value (CLV), 150
 - customer segments, Lean Canvas, 75
 - customers
 - CAB, 216
 - CAC, 150
 - CES, 142, 144
 - CLV, 150
 - distance, 165
 - ICP, 216
 - interviews, product discovery, 215–216
 - interviews, value creation, 49
 - journeys, 102–104
 - Lean Canvas, 75
 - metrics, 150
 - retention rates, 149
 - cycles, product delivery, 127–129
- ## D
-
- Dalio, Ray, 5
 - Dalmijn, Maarten
 - alignment of goals, 127–128
 - discovery process, 192
 - data metrics (actionable), 139
 - activation metrics, 150
 - activation rates, 150
 - AI insights, 149
 - alerts, 149
 - average session duration, 149
 - bounce rates, 149
 - CAC, 150
 - CES, 142, 144
 - churn rates, 150
 - CLV, 150
 - conversion rates, 149
 - CTR, 149
 - customer retention rates, 149
 - data traps, 140–141
 - defining, 144–146
 - engagement metrics, 149

- feature adoptions, 150
- gross margins, 150
- identifying, 142
- input/output metrics, 143–144
- laggard metrics, 141–143, 151
- leading metrics, 141–143, 151
- measuring metric results, 146–147
- monetary metrics, 150
- MRR, 150
- onboarding, 145
- onboarding completion rates, 150
- referral rates, 150
- return visits, 149
- scheduled reports, 148
- sharing metric results, 147–148
- sign-up rates, 150
- time saving techniques, 148–149
- unique visitors, 149
- vanity metrics, 140–141
- data traps, 140–141
- deadlines
 - fixed mindsets, 32
 - meeting, 32–33
 - mindsets, 32–33
 - strict deadlines, 32
- debt, tech
 - accelerated learning, 19–20
 - as investment, 131–133
 - product delivery, 131–133, 137, 223
 - risk/knowledge relationships, 132
- decision making, product strategies, 208
- decline phase, product life cycles, 69–70
- defining actionable metrics, 144–146
- Definition of Done (DoD), 131
- Definitions of Ready, 123–124
- delivery, product
 - alignment, 127–129, 134–135
 - backlogs, 121–125, 137, 226–227
 - clutter, avoiding, 223–224
 - cycles, 127–129
 - distractions, 223–224
 - DoD, 131
 - elements of, 120
 - excessive delivery, 59
 - failure, learning from, 224–226
 - feature factories, 120–121
 - feature removal, 223–224
 - health checks, 222
 - implementing, 222–223
 - Instagram, birth of, 225
 - investments, 131–133
 - meaningful delivery, 61–62
 - outcomes, measuring, 223
 - refinement sessions, 125–126
 - scalability, 223
 - sustainability, 221–222
 - team improvements, 135–136
 - tech debt, 131–133, 137, 223
 - Twitter (X), birth of, 226
 - workflows, 129–131
 - wrong approaches to, 120–121
- departments, feature factories, 8–9
- dependent teams, 154–155, 156–157
- design phase, coordinative development flows, 13
- desirability, idea evaluation, 17–18
- development phase, coordinative development flows, 14
- Diffusion of Innovations*, 67
- discovery, product, 89–90, 118, 213
 - assumptions, 216–218
 - B2B scenarios, 216
 - B2C scenarios, 215
 - biased product discovery, 91–92, 102
 - clarity of importance, 214–215
 - customer interviews, 215–216
 - excessive delivery, 59
 - excessive discovery, 59
 - health checks, 214
 - ideation, 218–219
 - insight creation, 215–216
 - mindful product discovery, 91, 102
 - overconfidence, 219
 - sustainable balance, 60
 - value drivers, 95–98
 - value proposition, 214–215, 219
- discovery journeys, 118
 - assumptions, 110–114, 118
 - clarifying goals, 101–102
 - customer journeys, 102–104
 - defined, 99–100
 - evidence, 116–117
 - example of, 100
 - hypotheses, 111
 - ideation techniques, 109–110
 - opportunity solution trees, 106–107, 108–109
 - product experiments, 114–117
 - reinventing the future, 109–110
 - success, defining, 101–102
 - user story mapping, 102–103, 112
 - value drivers, 104–109
- discovery process, product principles, 192
- distance, customer, 165
- distractions
 - product delivery, 223–224
 - product principles, 191
- doable but undesirable, 160
- DoD (Definition of Done), 131
- Dorsey, Jack, 226
- drivers, value creation, 38
- Driving Value with Sprint Goals: Humble Plans, Exceptional Results*, 127–128, 192
- dropping ideas, 17–20
- dual-track agile approaches, empowered teams, 158–159
- due dates, backlogs, 124
- Dweck, Carol, 23

dynamics, company, 197
 accountability, 199
 health checks, 198
 leadership, 198–199
 product leadership, 201–202
 product perception, 200
 risk tolerance, 203–204
 team topologies, 202

E

El Bulli, 56–61, 230
 empowered teams, 154–155, 169
 defined, 11–12
 dual-track agile approaches, 158–159
 obstacles, 158–159
 empowered-but teams, 154–155
 engagement metrics, 149
 escalating commitments, 92–93, 118
 ethics, idea evaluation, 17–18
 evaluation phase, collaborative development flows,
 17–18
 evidence
 and opinions, fixed mindsets, 28–30
 product experiments, 116–117
 product principles, 195
 evolving product principles, 190
 excessive delivery, product discovery, 59
 exchanges, in-person, 180
 experimentation phase, collaborative development flows,
 19–20
 experiments, product
 defining, 114–116
 evidence, 116
 template example, 115
 extensive product backlogs, 40–41

F

face-to-face engagement, 180–181
 failures
 avoidance, value creation, 50
 learning from (product delivery), 224–226
 rates, ideas, 15–16
 feasibility, idea evaluation, 17–19
 Feature Adoption Report (2019), 49
 feature adoptions, 49, 150
 feature factories, 1, 154–155
 building unnecessary features, 8–9
 company departments, 8–9
 operation of, 8–12
 product delivery, 120–121
 team/stakeholder collaborations, 8–9
 feature removal, product delivery, 223–224

feature reports, 49
 features (new)
 releasing, 10–12
 unnecessary features, 10–12
 feedback
 refinement sessions, 125
 sending, 231
 finishing over starting, product principles, 192–193
 fixed mindsets, 35
 deadlines, 32
 defined, 23–25
 failure avoidance, 50
 opinions over evidence, 28–30
 “outcome over output” principle, 27–28
 Frankfurt, Henry, 38
 freemium business models, 73

G

goals
 alignment, product delivery, 127–128
 clarifying, discovery journeys, 101–102
 mindsets, 32–33
 product principles, 191
 product strategies, 208–210
 good enough, product principles, 195
 good leaders, company dynamics, 198–199
 good product vision, 71
 gross margins, 150
 growth mindsets
 defined, 23–25
 failure avoidance, 50
 growth phase, product life cycles, 69–70

H

health checks
 company dynamics, 198
 product delivery, 222
 product discovery, 214
 product strategies, 206
 HiPPO (Highest Paid Person Opinions), 48–49
 Huwatscheck, Marion, 111
 hypotheses, 111

I

ICP (Ideal Customer Profiles), 216
 ideas
 dropping, 17–20
 evaluating, 17–20
 failure rates, 15–16
 pursuing, 17–18
 reality and, 19

ideation techniques
 concierge-driven ideation technique, 110
 crazy eights ideation technique, 109
 discovery journeys, 109–110
 product discovery, 218–219
 session-based ideation, 109
 web platforms, 110

identifying
 actionable metrics, 142
 assumptions, 111–113, 216–217
 target audiences, 66–68

IDEO, Innovation Trinity, 18

ignore cluster, stakeholder matrix, 174

ignoring assumptions, 114

impact of work (product principles), measuring, 193

implementing, product delivery, 222–223

importance of balance, 50

importance (product discovery), clarity of, 214–215

improving teams, product delivery, 135–136

independent teams, 154–155, 157–158

Innovation Trinity, 18

in-person exchanges, 180

input/output metrics, 143–144

insight creation, product discovery, 215–216

Instagram, birth of, 225

interactions, product principles, 188–189

interviews
 customer interviews, product discovery, 215–216
 observing, value creation, 49

introduction phase, product life cycles, 69–70

investments, tech debt as, 131–133

inviability, attractive but, 160

iPhones, product vision, 66

item formats, backlogs, 123–124

iTunes, 226

K

keep informed cluster, stakeholder matrix, 174

keep involved cluster, stakeholder matrix, 174

keep satisfied cluster, stakeholder matrix, 174

Kennedy, John F., 69–70

key metrics, Lean Canvas, 77

Kim, W. Chan, 80

knowledge/risk relationships, tech debt investments, 132

Krieger, Mike, 225

L

laggard metrics, 141–143, 151

launch phase
 collaborative development flows, 20–21
 coordinative development flows, 14–15

leadership
 accountability, 199
 company dynamics, 198–199
 product leadership, 201–202

leading metrics, 141–143, 151

Lean Analytics, 149

Lean Canvas, 66, 72–73
 channels, 76
 cost structures, 77
 crafting, 74–79
 customer segments, 75
 key metrics, 77
 Netflix, 79
 problems, defining, 75–76
 revenue streams, 77
 Slack, 79
 solutions, 76
 Substack, 75–79
 unfair advantages, 78–79
 unique value propositions, 76

lean product backlogs, 123

leap of faith assumptions, 114

learning
 building to learn, then to scale (product principles), 194
 from failure, product delivery, 224–226
 iteration, collaborative development flows, 17–19

lessons (overview), untrapping, 233–234

life cycles
 adoption life cycles, 67–68
 product life cycles, 69–70

lifespans, product strategies, 58–59

livelihood of product principles, maintaining, 188–189

Longman Dictionary of Contemporary English, 111

M

managing backlogs, 121–122

mapping user stories, 102–103, 112

marathons, meeting, 45–46

marketplace-based business models, 74

Maslow, Abraham H., 183

maturity phase, product life cycles, 69–70

Mauborgne, Renée, 80

Maurya, Ash
 Lean Canvas, 72, 74
Running Lean, 116

meaningful product delivery, 61–62

measuring
 metric results, 146–147
 outcomes, product delivery, 223

meetings
 marathons, value creation, 45–46
 structured meetings, communication, 177

Mendelow, Aubrey L., 173–175

metrics, actionable, 139

- activation metrics, 150
- activation rates, 150
- AI insights, 149
- alerts, 149
- average session duration, 149
- bounce rates, 149
- CAC, 150
- CES, 142, 144
- churn rates, 150
- CLV, 150
- conversion rates, 149
- CTR, 149
- customer retention rates, 149
- data traps, 140–141
- defining, 144–146
- engagement metrics, 149
- feature adoptions, 150
- gross margins, 150
- identifying, 142
- input/output metrics, 143–144
- laggard metrics, 141–143, 151
- leading metrics, 141–143, 151
- measuring metric results, 146–147
- monetary metrics, 150
- MRR, 150
- onboarding, 145
- onboarding completion rates, 150
- referral rates, 150
- return visits, 149
- scheduled reports, 148
- sharing metric results, 147–148
- sign-up rates, 150
- time saving techniques, 148–149
- unique visitors, 149
- vanity metrics, 140–141

metrics (key), Lean Canvas, 77

Microsoft Surface, product vision, 71

mindful product discovery, 91, 102

Mindset: The New Psychology of Success, 23

mindsets

- deadlines, 32–33
- defined, 23
- fixed mindsets, 23–30, 35, 50
- goals, 32–33
- growth mindsets, 23–25, 50
- impacts of, 26–27
- prescriptive roadmaps, 30–32, 35
- reality check surveys, 26–27
- solutions versus problem-solving, 33–34

monetary metrics, 150

Moore, Geoffrey, 68

MRR (Monthly Recurrent Revenue), 150

N

Netflix, Lean Canvas, 79

new features, releasing, 10–12

new products, backlogs, 122–123

newsletters, author's newsletter, 231

"no," saying, 46–47

Now-Next-Later roadmaps, 82–85

O

observing interviews, value creation, 49

Odeo, 226

OKR (Objectives and Key Results), 82

onboarding, 145

onboarding completion rates, 150

On Bullshit, 38

on-demand business models, 73

online/author's resources

- blog, 3
- feedback, sending, 231
- Lean Canvas, 74
- product adoption life cycles, 68
- roadmap templates, 83
- stakeholder matrix, 173
- Substack, 3
- updates, 231

opinion-driven decisions

- HiPPO, 48–49
- value creation, 48–49

opinions over evidence, fixed mindsets, 28–30

opportunity solution trees, 106–107

- assumptions, 110
- prioritizing, 108–109

Osterwalder, Alexander

- Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, 74
- importance of balance, 50

outcomes

- measuring outcomes, product delivery, 223
- "outcome over output" principle, fixed mindsets, 27–28

output

- approval, value creation, 44–45
- metrics, 143–144
- reports, 42
- roadmaps, 143

overconfidence, product discovery, 219

owners, product, 164–165

ownership and accountability, 206–207

P

partnering over serving, product strategies, 207

passengers, value creation, 38

- Patton, Jeff, 39, 102–103
- perception of product, 200
- Pereira, David, author resources
 - blog, 3
 - feedback, sending, 231
 - Lean Canvas, 74
 - product adoption life cycles, 68
 - roadmap templates, 83
 - stakeholder matrix, 173
 - Substack, 3
- Pigneus, Yves, 74
- pixel-perfect designers, 167
- plotting Value Curves, 81–82
- prescriptive roadmaps, 30–32, 35
- principles, product, 185, 196
 - A even over B, 188
 - A instead of B, 188
 - A over B, 188
 - alignment, 194
 - building to learn, then to scale, 194
 - collaboration, 192
 - crafting, 186–188
 - discovery process, 192
 - distractions, 191
 - evidence over opinion, 195
 - evolving, 190
 - finishing over starting, 192–193
 - goals, 191
 - good enough, 195
 - impact of work, measuring, 193
 - interactions, 188–189
 - livelihood of principles, maintaining, 188–189
 - problem solving, 195
 - risk taking, 193
 - simplicity, 194–195
 - statements, 188
 - usage, assessing, 189
- prioritizing
 - assumptions, 113–114
 - coordinative development flows, 13
 - product strategies, 207
 - roadmaps, 86
 - value drivers, 107–109
- problem-solving
 - product principles, 195
 - solutions versus, mindsets, 33–34
- problems (Lean Canvas), defining, 75–76
- product adoption life cycles, 67–68
- product backlogs
 - product delivery, 226–227
 - value creation, 40–41
- product delivery
 - alignment, 127–129, 134–135
 - backlogs, 121–125, 137, 226–227
 - clutter, avoiding, 223–224
 - cycles, 127–129
 - distractions, 223–224
 - DoD, 131
 - elements of, 120
 - failure, learning from, 224–226
 - feature factories, 120–121
 - feature removal, 223–224
 - health checks, 222
 - implementing, 222–223
 - Instagram, birth of, 225
 - investments, 131–133
 - meaningful delivery, 61–62
 - outcomes, measuring, 223
 - refinement sessions, 125–126
 - scalability, 223
 - sustainability, 221–222
 - team improvements, 135–136
 - tech debt, 131–133, 137, 223
 - Twitter (X), birth of, 226
 - workflows, 129–131
 - wrong approaches to, 120–121
- product designers, 161, 167, 169
- product development
 - collaborative development flows, 12, 16–21, 21
 - coordinative development flows, 12, 12–16, 21
- Product Direction: How to Build Successful Products at Scale with Strategy, Roadmaps, and Objectives and Key Results, 65*
- product discovery, 89–90, 118, 213
 - assumptions, 216–218
 - B2B scenarios, 216
 - B2C scenarios, 215
 - biased product discovery, 91–92, 102
 - clarity of importance, 214–215
 - customer interviews, 215–216
 - excessive delivery, 59
 - excessive discovery, 59
 - health checks, 214
 - ideation, 218–219
 - insight creation, 215–216
 - mindful product discovery, 91, 102
 - overconfidence, 219
 - sustainable balance, 60
 - value drivers, 95–98
 - value proposition, 214–215, 219
- product experiments
 - defining, 114–116
 - evidence, 116
 - template example, 115
- product leadership, 201–202
- product life cycles, 69–70
- product management
 - finding the right ingredients for a situation, 62–63
 - key ingredients, 54
 - meaningful product delivery, 61–62
 - product delivery, 61–62
 - product discovery, 59–60
 - product strategies, 57–59, 65–87
- product managers, 161, 162–164, 169
- product owners, 164–165

- product perception, 200
 - product principles, 185, 196
 - A even over B, 188
 - A instead of B, 188
 - A over B, 188
 - alignment, 194
 - building to learn, then to scale, 194
 - collaboration, 192
 - crafting, 186–188
 - discovery process, 192
 - distractions, 191
 - evidence over opinion, 195
 - evolving, 190
 - finishing over starting, 192–193
 - goals, 191
 - good enough, 195
 - impact of work, measuring, 193
 - interactions, 188–189
 - livelihood of principles, maintaining, 188–189
 - problem solving, 195
 - risk taking, 193
 - simplicity, 194–195
 - statements, 188
 - usage, assessing, 188–189
 - product strategies, 57, 65, 87, 205
 - accountability, 211
 - analysis paralysis, 211
 - clarifying, 58
 - clarity of goals, 208–210
 - collaboration, 210–211
 - context switching, 209–210
 - decision making, 208
 - goals, 208–210
 - health checks, 206
 - implementing, 206
 - key elements, 66
 - Lean Canvas, 66, 72–79
 - lifespans, 58–59
 - ownership and accountability, 206–207
 - partnering over serving, 207
 - prioritization, 207
 - product adoption life cycles, 67–68
 - product life cycles, 69–70
 - product vision, 66, 69–72
 - progress over analysis, 208
 - roadmaps, 66, 82–86
 - target audiences, identifying, 66–68
 - Value Curves, 66, 80–82
 - product teams, 161, 169
 - backlogs, 10
 - business stakeholder collaborations, 9–10
 - common setups, 154–155
 - product Venn diagrams, 18, 160–161
 - product vision, 66
 - bad product vision, 70–71
 - compelling visions, 69–70
 - crafting, 71–72
 - good product vision, 71
 - iPhones, 66
 - Surface, 71
 - progress
 - collaboration, sharing progress, 176–177
 - over analysis, product strategies, 208
 - promising but unfeasible, 160
 - pursuing ideas, 17–18
- R**
- random 1:1 interactions, 180
 - Ready, Definitions of, 123–124
 - reality and ideas, 19
 - reality check surveys, 26–27
 - referral rates, 150
 - refinement sessions, 125–126
 - reinventing the future, discovery journeys, 109–110
 - relationships, building, 171
 - alignment, 177–179
 - authenticity, 180
 - consistency, 175
 - face-to-face engagement, 180–181
 - in-person exchanges, 180
 - random 1:1 interactions, 180
 - shared understanding, 175–177
 - sharing progress, 176–177
 - spontaneity, 180
 - stakeholders, 172–181
 - structured meetings, 177
 - transparency, 176
 - releasing new features, 10–12
 - removing features, product delivery, 223–224
 - reports
 - feature reports, 49
 - output reports, 42
 - scheduled reports, 148
 - requirements (value creation) gathering, 39–40
 - resources, online/author's
 - blog, 3
 - feedback, sending, 231
 - Lean Canvas, 74
 - product adoption life cycles, 68
 - roadmap templates, 83
 - stakeholder matrix, 173
 - Substack, 3
 - updates, 231
 - responsibilities/roles in teams, 160–162, 164
 - results (metrics)
 - measuring, 146–147
 - sharing, 147–148
 - retention rates, customer, 149
 - return visits, 149
 - revenue streams, Lean Canvas, 77

- risk
 - taking, product principles, 193
 - tolerance, company dynamics, 203–204
 - risk/knowledge relationships, tech debt investments, 132
 - roadmaps
 - example of, 85
 - laggard metrics, 142–143
 - Now-Next-Later roadmaps, 82–85
 - OKR, 82
 - output roadmaps, 143
 - prescriptive roadmaps, 30–32, 35
 - prioritization, 86
 - product strategies, 66, 82–86
 - templates, 82–83
 - trash column, 82–83
 - Rogers, Everett M.67
 - roles/responsibilities in teams, 160–162, 164
 - Rotter, Julian B.8
 - Running Lean*, 116
 - running products, backlogs, 123
- S**
- SaaS (Software as a Service), 73
 - saving timer, actionable metrics, 148–149
 - saying “no” strategies, value creation, 46–47
 - scalability
 - building to learn, then to scale (product principles), 194
 - product delivery, 223
 - scheduled reports, 148
 - Scrum, output reports, 42
 - Scrum Guide 2017, The*, 126
 - Scrum Guide 2020, The*, 164
 - Scrum Masters, output reports, 42
 - services, SaaS, 73
 - serving (product strategies), partnering over, 207
 - sessions
 - average duration, 149
 - ideation, 109
 - setups, teams
 - accountability, 161–162
 - collaborative teams, 154–155, 159
 - common setups, 154–155
 - coordinative teams, 154–155
 - dependent teams, 154–155, 156–157
 - empowered teams, 154–155, 158–159, 169
 - empowered-but teams, 154–155
 - feature factories, 154–155
 - independent teams, 154–155, 157–158
 - roles/responsibilities, 160–162
 - size of, 159
 - trapped teams, 154–155
 - shared understanding
 - consistency, 175
 - sharing progress, 176–177
 - structured meetings, 177
 - transparency, 176
 - sharing
 - progress, collaboration, 176–177
 - metric results, 147–148
 - sign-up rates, 150
 - silos, 161–162
 - simplicity, product principles, 194–195
 - Sinek, Simon
 - crafting product vision, 71
 - interactions, 179
 - “Why We Form Tribes”, 179
 - size of teams, determining, 159
 - Slack, Lean Canvas, 79
 - Software as a Service (SaaS), 73
 - software engineers, 161, 165–167, 169
 - solutions
 - Lean Canvas, 76
 - problem-solving versus solutions, mindsets, 33–34
 - solving problems, product principles, 195
 - spontaneity, communication/collaboration, 180
 - stakeholders
 - alignment, 177–179
 - building relationships, 172–181
 - collaboration, 173–175
 - defined, 172
 - matrix, 173–175
 - Start With Why*, 71
 - statements, product principles, 188
 - status-updating micro-blogging, 226
 - Stone, Biz, 226
 - story mapping, user, 39, 102–103, 112
 - strategies, product, 57, 65, 87, 205
 - accountability, 211
 - analysis paralysis, 211
 - clarifying, 58
 - clarity of goals, 208–210
 - collaboration, 210–211
 - context switching, 209–210
 - decision making, 208
 - goals, 208–210
 - health checks, 206
 - implementing, 206
 - key elements, 66
 - Lean Canvas, 66, 72–79
 - lifespans, 58–59
 - ownership and accountability, 206–207
 - partnering over serving, 207
 - prioritization, 207
 - product adoption life cycles, 67–68
 - product life cycles, 69–70
 - product vision, 66, 69–72
 - progress over analysis, 208
 - roadmaps, 66, 82–86
 - target audiences, identifying, 66–68
 - Value Curves, 66, 80–82

strict deadlines, 32
 structured meetings, communication, 177
 structuring value drivers, 106–107
 subscription-based business models, 73
 Substack
 author's substack, 3
 Lean Canvas, 75–79
 success (discovery journeys), defining, 101–102
 sunk-cost fallacy, 93–94
 Surface (Microsoft), product vision, 71
 sustainability
 balance, product discovery, 60
 product delivery, 221–222
 sweet spots, 160
 switching context, product strategies, 209–210
 Systrom, Kevin, 225

T

target audiences, identifying, 66–68
 teams
 accountability, 44–45, 161–162
 Agile Coaches, 168–169
 agile rulers, 168
 agile teams, 42
 backlog managers, 162–164
 coders, 165–167
 collaborative teams, 154–155, 159
 common setups, 154–155
 communication, 165
 consensus, 162
 coordinative teams, 154–155
 customer distance, 165
 dependent teams, 154–155, 156–157
 empowered teams, 11–12, 154–155, 158–159, 169
 empowered-but teams, 154–155
 feature factories, 154–155
 improving, product delivery, 135–136
 independent teams, 154–155, 157–158
 pixel-perfect designers, 167
 product designers, 161, 167, 169
 product managers, 161, 162–164, 169
 product owners, 164–165
 product teams, 9–10, 154–155, 161, 169
 roles/responsibilities, 160–162, 164
 silos, 161–162
 size of, 159
 software engineers, 161, 165–167, 169
 topologies, company dynamics, 202
 trapped teams, 154–155
 tech debt
 accelerated learning, 19–20
 as investment, 131–133
 product delivery, 131–133, 137, 223
 risk/knowledge relationships, 132

templates
 product experiments, 115
 roadmaps, 82–83
 testing
 assumptions, 118, 218
 coordinative development flows, 14
Testing Business Ideas, 19
 time saving techniques, actionable metrics, 148–149
 topologies of teams, company dynamics, 202
 Torres, Teresa
 Continuous Discovery Habits, 103–104
 opportunity solution trees, 106–107
Toward a Psychology of Being, 183
 transparency, communication, 176
 trapped teams, 154–155
 trash column, roadmaps, 82–83
 Twitter (X), birth of, 226

U

uncovering value drivers, 104–106
 understanding, shared
 consistency, 175
 sharing progress, 176–177
 structured meetings, 177
 transparency, 176
 undesirable, doable but, 160
 unethical, valuable but, 160
 unfair advantages, Lean Canvas, 78–79
 unfeasible, promising but, 160
 unique value propositions, Lean Canvas, 76
 unique visitors, 149
 unnecessary features, 10–12
 untrapping lessons (overview), 233–234
 updates, online resources, 231
 usage of product principles, assessing, 189
 user story mapping, 39, 102–103, 112
User Story Mapping, 39, 102–103
 user tests, coordinative development flows, 13

V

valuable but unethical, 160
 value, CLV, 150
 value creation, 37
 accountability, 44–45
 backlogs, 121–122
 balance, importance of, 50
 bottlenecks, 47–48
 bridge communication, 47–48
 “bullshit management”, 38–39, 51–52
 consensus-driven decisions, 43–44
 drivers, 38
 failure avoidance, 50

- feature reports, 49
- meeting marathons, 45–46
- observing interviews, 49
- opinion-driven decisions, 48–49
- output approval, 44–45
- output reports, 42
- passengers, 38
- product backlogs, 40–41
- requirements gathering, 39–40
- saying “no” strategies, 46–47
- Value Curves, 66
 - Blue Ocean strategy, 80–82
 - creating, 80
 - example of, 81
 - plotting, 81–82
- value drivers, 95–98
 - opportunity solution trees, 106–107, 108–109
 - prioritizing, 107–109
 - structuring, 106–107
 - uncovering, 104–106
- value proposition
 - Lean Canvas, 76
 - product discovery, 214–215, 219
 - unique value propositions, 76
- vanity metrics, 140–141
- velocity
 - agile teams, 42
 - defined, 42
- Venn diagrams, product, 18

- Vernon, Raymond, 69–70
- Verveen, Jos, 38
- viability, idea evaluation, 17–19
- victim of the circumstance, 8
- vision, product, 66
 - bad product vision, 70–71
 - compelling visions, 69–70
 - crafting, 71–72
 - good product vision, 71
 - iPhones, 66
 - Surface, 71
- visitors, unique, 149
- visits, return, 149

W

- waste, 160
- web platforms, 110
- “Why We Form Tribes”, 179
- workflows, product delivery, 129–131
- Working Backwards*, 143

X - Y - Z

- X (formerly Twitter), birth of, 226
- Yoskovitz, Benjamin, *Lean Analytics*, 149