

# **TURN IDEAS INTO PRODUCTS:**

**A Playbook for Defining and  
Delivering Technology  
Products**

Steve Johnson

*Turn Ideas Into Products: A Playbook for Defining and Delivering Technology Products*, copyright © 2020 Under10 Consulting LLC

Some text previously published in *Look Beyond The Product: The Business of Agile Product Management*, copyright © 2014 Under10 Consulting LLC

Cover image by Gürcan GÜR from Pixabay

All rights reserved, no part of this publication may be copied, reproduced, or stored by any means, electronic, mechanical, optical, or otherwise without the prior consent of the author.

200303

## What people are saying

Written in a concise and conversational tone, *Turn Ideas Into Products* is full of practical tools and wisdom. If you want to understand the core of effective product management and focus on the few things that matter the most, this is the book for you.

Eleanor K, life sciences  
product strategist and marketer

\*

In *Turn Ideas Into Products*, Steve Johnson offers a simple but powerful approach to defining and delivering technology products. I've read his works for years and this is the best thing he's written so far.

Barbara N, expert in  
product management and marketing

\*

This book provides a practical way for product managers to juggle the competing priorities of the job. Its brevity is refreshing, yet it covers the breadth of work that product managers must tackle to have significant business impact. Most product managers naturally focus on a few areas of the job while neglecting some of the others. This book will help product managers maintain balance among all of their important responsibilities. Excellent for both experienced and new product managers leading products in any phase of the life cycle, from newly launched to mature products.

Felicia A, VP of product  
management and product ops

\*

*Turn Ideas Into Products* is the process-simplification guide that every product team needs. In the book, Steve

clearly lays out the essential tools that product leaders need to design, build and ship product that solves the real problems of real buyers. How refreshing. Plus, Steve gets Win-Loss like few others.

Charles Topping,  
The Win-Loss Agency

\*

*Turning Ideas Into Products* is a concise and impactful reference book for anyone working on products. This book is a prime example of why Steve is called “The Godfather of Product Management.”

Alicia D, product management leader

\*

Steve could have written a 300-page manual or handbook — he has the decades of experience, knowledge and skill to do that. Instead, Steve distilled his expertise and wisdom into an elegant, clearly written, logical package. This short blueprint is like a customizable home plan, a guide with multiple options to reconfigure and try until you find the perfect fit for your particular context — organization, culture, product, team, and market. I wish I’d had this gem 15 years ago.

Jason Tanner, CEO,  
Applied Frameworks

**CONTENTS**

**FOREWORD by Rich Mironov ..... 1**

**Introduction: Your Product Adventure..... 3**

    Broken Processes, Broken Dreams ..... 5

    You Need a Process Framework ..... 7

    QuartzOpen: An Innovation Framework..... 8

    The Basic Building Blocks..... 12

    Adventure: Meghan the Product Manager ..... 15

    Adventure: Tom the Entrepreneur..... 17

**Chapter 1: Discover ..... 21**

    Listen and Observe..... 22

    Define Your Personas..... 25

    Getting Market Insights..... 34

    Problem Discovery: Interviews and Observation..... 35

    Solution Validation: Experiments and Surveys ..... 37

    Opportunity Scoring..... 40

**Chapter 2: Commit ..... 55**

    Define Your Product ..... 56

    Identify the Competitors..... 58

    Pricing and Revenue Forecasting..... 59

    Allocate a Product-Specific Development Budget..... 62

    Allocate a Product-Specific Promotion Budget ..... 64

    Build a Business Justification ..... 65

    Gain Executive Commitment..... 66

**Chapter 3: Describe ..... 69**

    Persona Problems are Product Stories..... 70

Define Features and Capabilities.....	72
Describe Needs, Constraints, and Outcomes .....	76
The Problem of Story Detail .....	78
<b>Chapter 4: Create.....</b>	<b>81</b>
Product Team Productivity.....	82
Inspire the Product Team .....	84
Feed Your Team Using “Pull” .....	85
Prioritize for Business Value.....	87
Product Stories and The Five Queues .....	89
Formalize Story Acceptance with a Demo.....	92
The Five Queues Determine the Product Roadmap .....	95
Can You Develop Faster Than You Sell? .....	97
Protect Your Product Team.....	98
<b>Chapter 5: Deliver .....</b>	<b>101</b>
Product Marketing and The Promotion Team.....	102
Separate Release from Launch.....	103
A Compelling Release.....	106
Prepare for Launch .....	107
Align Product Launch with Market Rhythms .....	110
A Phased Approach.....	112
Launch Plans and The Four Queues.....	117
Product Leaders and Launch Plans .....	119
<b>Chapter 6: Connect.....</b>	<b>121</b>
Create a Sales Playbook .....	123
The Importance of Sales Engineering .....	125
Win-Loss Analysis .....	126
Diagnose the Buyer’s Journey.....	129

Managing for Customer Success.....	131
<b>Chapter 7: Learn.....</b>	<b>135</b>
Market Retrospective .....	137
Business Retrospective.....	138
Process Retrospectives .....	139
Always Be Learning.....	140
<b>Chapter 8: On Product Leadership.....</b>	<b>143</b>
Defining Departmental Roles.....	146
Roles Through the Product Life Cycle.....	148
Three Roles of Product.....	152
The Unicorn and the Purple Squirrel.....	155
What Makes a Product Leader? .....	156
Your First Days as a Product Leader .....	159
Standardize Your Team with Product Ops.....	164
<b>About the Author.....</b>	<b>169</b>





## FOREWORD BY RICH MIRONOV

Steve Johnson has been mentoring product managers and product leaders for decades: hundreds of thousands of product folks have read his blogs, attended his workshops, used his templates, and laughed at the emotional truths of his war stories.

Steve has helped establish our worldwide product management tribe.

In *Turn Ideas Into Products*, Steve leads us step-by-step through validating, describing, building, launching, and improving technology products. This book delivers hard-won wisdom and practical advice without falling into obscure terminology or rigid one-size-fits-all methodologies.

It provides a broad outline of the product journey—highlighting useful tools and identifying typical traps along the way—so that we can each deliver the unique tech products we’ve been thinking about.

Both novices and seasoned product leaders will find important lessons here.

Rich Mironov

CEO of Mironov Consulting, founder of Product Camp, and author of *The Art of Product Management*

# INTRODUCTION: YOUR PRODUCT ADVENTURE

*“Products often get released through individual heroics, despite the company’s best efforts to prevent it.”—Steve Johnson, author, product strategist*

We’ve all heard stories of amazing product successes: the brilliant college kid who started a business in his dorm room; the team who built a business from the back of a napkin with just a few friends and sold it for millions. Yet for every amazing success story, there are thousands of stories of products that went nowhere.

Most of us are not looking at billion-dollar valuations; we’re not even looking for an exit. Instead we have a few ideas—some innovative, some not—and we’re trying to determine which to pursue. Likely, you’re working for a

company today and you need a step-by-step approach to turn ideas, regardless of their source, into business outcomes.

One company acquired a product that was seemingly successful in Europe; the company leaders believed it would be even more successful here in the United States. Alas, discussions with both customers and colleagues revealed the product would not do well for many reasons—at least, not from this company. A business-oriented product manager met with the company leaders and revealed the customer issues as well as the internal issues. He recommended they kill the product.

And surprisingly, they did.

Because he presented them with market facts.

Another company had an idea for a product. They built a prototype, shared it with some folks who were the ideal clients, made some revisions, and shared it with some more potential buyers. By the time the product was ready for introduction, they knew they had the right features and the right message.

Because they had market facts.

So can you.

My approach uses market insights to create a queue of product ideas focused on problems to be solved. Each idea is prioritized for business value. You build a team to create the solution and give them the most important items when they're ready to accept more work. You deliver new capabilities to market when customers are ready to receive them.

However, when a product failure occurs—as it often does for tech companies—the natural inclination is to develop a formal product process.

And that makes sense.

(Or does it?)

## Broken Processes, Broken Dreams

*“A bad system will beat a good person every time.”—W Edwards Deming, American engineer, statistician, professor, author, lecturer, and management consultant*

People who design internal processes have the best of intentions. They want to ensure all stakeholders are engaged and that the results are in line with the company’s objectives. But despite their best intentions, they often deliver an overbearing process that requires company-wide participation and assumes perfect information.

You’ve seen it. Processes that mandate buy-in from every department. Processes that require unrealistic precision in cost and revenue estimates. Processes that assume perfect information.

It begins simply enough.

In the early days, we agreed to deliver a certain set of functionality. The company leaders were all in the room when we made the decisions. Yet, as with any project, we defined some pretty aggressive goals. We believed the requirements were well-understood and locked down, and we assumed the team would work 40 hours every week without distractions (or sick days or vacations).

But somehow the project was delayed.

And delayed again.

And when the project was finally delivered, it fell well short of expectations.

It seems that the team delivered most of what had been agreed upon, but some key capabilities were incomplete or missing altogether.

Of course, a lot had happened since those early decisions. An exec had a new idea, a sales guy needed a feature to close a deal, one of the lead developers left for a new opportunity, the bug count was higher than expected, and everyone had significantly under-estimated the effort and over-estimated the product team's productivity.

Maybe the process needs more structure, more rigor.

So someone said, "Why don't we create a marketing requirements document?" And that seemed like a good idea. The next time we started a project, we created a long marketing requirements document (MRD) that defined the market, the features, the schedule, and a bunch of other stuff. And everyone seemed aligned.

But on delivery, a VP said, "Hey, wait a minute. Where's the feature I wanted?" Alas, it got dropped because it wasn't called out specifically in the MRD.

And someone said, "At my last company, we had a product requirements document. A PRD lists every feature we plan to deliver." And that seemed like a good idea. For the next project, we wrote an MRD and a PRD. We built according to these documents and delivered the next release.

But on delivery, a VP said, “We ran into a big architecture issue. Why didn’t you address it?”

And someone said, “You know, I’ve heard a lot of companies have started using a functional specification document. An FSD describes the full implementation of every feature so we don’t have these problems in the future.” And that seemed like a good idea.

So now, before we can begin anything, we need to produce an MRD, a PRD, plus an FSD? That’s crazy!

That’s when product teams—managers, marketers, designers, and developers—realized they were spending more time creating documents than they spend creating products.

Nowadays most companies have freed themselves from this old-school, document-heavy approach. We don’t need hundreds of pages, but we do need more than a drawing on a cocktail napkin.

## **You Need a Process Framework**

In my consulting work, I encounter one team after another with clumsy processes cobbled together from myriad sources that ultimately rely on team heroics to generate success.

The number of methods and frameworks for managing and marketing products is simply bewildering. Maybe your product team members have attended a training seminar, read the latest book, or watched a few video presentations. But there are so many conflicting definitions and so many different approaches. Perhaps that’s why people integrate multiple methods instead of following one method religiously. And that makes sense.

In the software world, many—perhaps most—product teams have adopted some form of agile development. In agile, small teams work for two weeks to create the most valuable parts of a product or solution, they share the solution with others, and gather feedback to incorporate into the next two weeks of work. Smart.

But these development methods are focused primarily on the creation process. What about what comes before and after? We need something similar for business planning and for go-to-market as well. Which solutions should we pursue and which should we not? Once we have a solution, what is the best way to deliver it to customers?

We need to extend the lessons of agile development to the entire product life cycle, from idea to market, from innovation to revenue.

Yes, you'll want to consider common practices when designing a planning process, but you'll need to adapt them to your unique issues. Your staffing. Your products. You need to find the ideal combination of process and documentation that fits your organization's need for information and risk mitigation.

## **QuartzOpen: An Innovation Framework**

Just as Scrum and other agile methods have optimized development projects, we need something to organize our thinking before, during, and after product creation. A way to guide you to build the right product and introduce that product to the market successfully.



The QuartzOpen Framework™ was developed to reflect that.

I was the original architect of a comprehensive planning process that became QuartzOpen. QuartzOpen is a product innovation methodology that provides both process and framework. Quartz links processes and tools to provide a consolidated view of the business at every stage of innovation.

QuartzOpen is light on handoffs and gated processes by design. This allows for smoother transitions between stages and teams. Being this nimble means speeding velocity, improving decision-time, and opening up more room for innovation.

QuartzOpen recognizes that cross-functional teams use their own processes and methodologies. Quartz was designed from the ground up to integrate well with others. It's a framework of frameworks, a method of methods.

Most planning methods take a “step 1, 2, 3” approach as if you're starting from nothing. They assume you don't already have a solution in some phase of its life cycle. However, the reality is this: most of us don't start from the beginning. We usually start somewhere in the middle.

The process for building products isn't a line, it's a wheel.

The key to the QuartzOpen Framework is its recursive hexagonal design. The needs of the entire organization are accounted for within six discrete facets organized into three phases of planning and execution.



In a typical scenario, we move high-priority ideas from DISCOVER to COMMIT when company leaders or investors allocate funds and dedicate resources to the idea. Some firms call this *product strategy*.

Once we have the go-ahead, we DESCRIBE and CREATE the product, working with a technical team that is assigned specifically to this product. In technology firms, this is generally called *product management*.

Along with building the product, we work on items to DELIVER the product and CONNECT with customers through our marketing and sales efforts. Sometimes we refer to this as *product marketing*.

The center of QuartzOpen is LEARN. That's because learning needs to be at the core of everything we do in business and continues to inform each phase in QuartzOpen.

LEARN includes idea discovery, continuous solution validation, and periodic retrospectives to ensure success. All

teams at all times should be actively listening to the market, to peers, to internal customers, and to cross-functional partners—iterating the planning and execution accordingly.

LEARN evokes the importance of keeping an open mind to new ideas throughout the process of bringing innovations to your business. Learning is necessary at every stage—in fact, learning is never completed. You continue to learn by analyzing the business results to determine what worked and what didn't so you can refine your idea and go through another cycle.

The QuartzOpen Framework has guided hundreds of companies implement a simple idea-to-market process. Ideally, with under 10 documents and under 10 steps. Each step requires one or more of the key artifacts—but only you can determine exactly what information you need for each step.

Here's the thing: Your company or organization is unique.

No one else offers exactly what you do with the people and resources you have. And that's the strength of the QuartzOpen approach. Adapt it to your business, your products, your processes, and your people.

At the highest level, QuartzOpen reveals the activities and artifacts necessary to *discover* problems to solve in the future, *describe* the next set of deliverables, and *deliver* growth for your current set of products.

## The Basic Building Blocks

*“It’s not easy, but it’s simple.” — Eric Thomas, motivational speaker and author of The Secret to Success*

Let’s keep it simple.

**Personas** represent people who buy and use your products; they serve as a stand-in for markets and segments. Using personas, you can easily size a market by estimating the number of folks who meet the persona definition.

Personas have **problems**, situations that keep them up at night or irritations they’d like to address. The thing is, people are really good at describing their problems; people are really bad at designing solutions. Don’t ask your customers what features they want; ask them about their problems.

Personas and problems are the basic building blocks of a product. Who is it for? Why do they need it? What are their pains? What are their goals? What will inspire them? Or even excite them? When you understand personas and problems, deciding what to deliver becomes much easier.

Many firms start with a technology or an invention and ask, “Who needs this?” I prefer to start with personas and their problems and ask, “How do we solve this?”

It’s easier to find products for people you know than to find people for products you know.

And I should mention this: when I say “product,” I’m not thinking “software,” although many do. **Products solve problems for personas.** We may solve those problems with

services, data, apps, software, hardware, or something else. Some firms refer to these as “solutions” or “offers” but I’ll use the term “product.”

In fact, some software and hardware aren’t really products until you add services to them. Services—such as implementation assistance, onsite installation, technical support, training—should be packaged so they are delivered consistently, just like software. That means you define the service so that it can be delivered again and again with minimal modification. You don’t want to be re-inventing the service each time you engage with a client, so service is product.

For example, there are myriad tools for creating and maintaining a website or blog. Many are actually more service than software. Website *products* offer easy hosting, ready-to-use templates and entire sample sites with better tools for composing, royalty-free photos, and 24x7 technical support. Very little of the website product is software; it’s mostly services.

I’ve simplified the product lifecycle process down to under ten artifacts. The term “artifact” is used for each of the basic building blocks of a product. An artifact is a template or worksheet that’s been filled with information you’ve discovered. There are artifacts for the personas, products, competitors, and so on.

These are the key elements that serve as the basic building blocks for today’s product playbook:

- Personas
- Problems
- Products
- Stories

- Releases and Launches (which are different as you'll learn later)

These key artifacts are the basis of all products. You'll want to create them if they don't already exist and then revisit them periodically to consider how they reflect your latest experiences and current thinking.

During the initial phase when you're describing your products and problems, you'll probably create one or two personas. Once you get into the later phases, you've learned more about the market and may have identified a few more relevant personas. Along the way, as you learn more, you'll revise your persona information.

The same is true for any of the core building blocks. You don't just create them and forget them.

I believe in a small number of living documents. The ones every product leader needs to master to ensure their product's success. The critical ones.

Only spend time on living documents—the ones you keep up-to-date and reference frequently.

Whether stored online or on printed pages, your product playbook is the set of artifacts you create; documents that are revised and reused throughout the product life cycle. You'll refine these documents over time. You'll make changes as you learn more about the market, their problems, and your solution. And you'll refer back to these to see just how far you've come.

But what's a typical scenario? Again, you're likely to join a product team at any step in this cycle—if that's your situation, just jump to the appropriate section of this book. If some terms are strange to you, keep reading—they will be explained.

## Adventure: Meghan the Product Manager

Poor Meghan! She's been promoted to product manager (whatever that means) for a company's flagship product. She has business skills and technology expertise, but it seems as if she's always rushing from one fire to the next. She's continually cleaning up the mistakes made by others.

Why would organizations hire a product manager mid-project? Here are some common scenarios:

- The former product manager has left for a new opportunity.
- The president has finally decided he or she needs to focus on the overall business and has hired a product manager to take over the product from day to day.
- The developers have been working for months without a product manager and now they've realized the need for a full-time person to guide product strategy.

Using QuartzOpen as a guide, Meghan has joined a company during the CREATE phase. She needs to keep the team on-task and protected from interference. However, her primary responsibility is preparing to launch the product.

What is Meghan's product adventure?

**CREATE.** Identify any barriers or distractions for the product team. Reverse-engineer the primary product stories from features already planned. Create or revise market and product personas. Monitor the status of product story development. Formalize story acceptance using Kanban queues (more on these concepts later).

**DELIVER.** Engage with potential customers to validate personas, product capabilities, and messaging. Brief the promotions team with successful implementations, market

feedback, and updated personas. Define one or more launch plans.

**CONNECT.** Monitor the status of launch plans using Kanban queues. Launch the product in phases: invitation, soft launch, and global launch. Use feedback from one phase to refine plans for the next phase. Support sales teams to identify necessary sales tools (such as an ROI calculator) and customer self-service deliverables (like a self-assessment worksheet).

**DISCOVER.** Conduct win-loss interviews and perform analysis for at least five accounts. Perform a retrospective on sales efforts and recommend revisions to sales tools. Perform a retrospective on revenue achievements and market share. Perform a retrospective on internal processes to identify areas to improve. Conduct opportunity scoring for the business model.

**COMMIT.** Update competitive profiles for existing and new competitors. Revisit development and promotions budgets. Refresh business documents and brief the executive team with revised business plan and product roadmap.

**DESCRIBE.** Brief the product team on successful implementations, market feedback, and updated personas. Rank planned product stories using Quick Prioritization. Conduct discovery sessions to provide the appropriate level of detail to the product team.

**CREATE.** Monitor the status of product story development using Kanban queues.

Meghan will continue this cycle again and again until the product gets retired or she gets promoted. The documentation



and artifacts—her product playbook—will help the next product manager get up to speed quickly.

Maybe you're like Meghan. Your onboarding was something like, "Welcome aboard. You're a product manager."

Soon there are product stories to accept, launch events to plan, and requests from salespeople to visit their top clients. You don't want to be the one who says, "Everybody just hold up until I can get organized." Instead, just dive in. Participate as best you can. And discover and document as you go.

## Adventure: Tom the Entrepreneur

Tom is great! He's a serial entrepreneur with an idea for a new product or service. He wants to evaluate his idea before he seeks outside investment, but Tom needs to back up a step to DISCOVER to make sure he's clear on the *who* and *what* of his idea. Who are his personas? What are their problems?

What is Tom's product adventure?

**DISCOVER.** Create a basic profile of the product idea. Interview at least five potential customers to understand their problems. Develop market and product personas. Engage with another five potential customers to validate personas and the proposed solution.

**COMMIT.** Create competitive profiles for existing and new competitors. Define development and promotions budgets. Create business justification with recommendations. Brief investors with business model and product roadmap.

**DESCRIBE.** Create a product team of developers, designers, and testers. Brief the product team with market feedback and updated personas. Rank planned product stories using Quick Prioritization. Conduct discovery sessions to provide the appropriate level of detail to the product team.

**CREATE.** Monitor the status of product story development using Kanban queues.

**DELIVER.** Revise product personas and positioning based on market feedback. Engage with potential customers to validate personas, product capabilities, and messaging. Create a promotions team and brief them on successful implementations, market feedback, and updated personas. Define one or more launch plans.

**CONNECT.** Monitor the status of launch plans using Kanban queues. Launch the product in phases: invitation, soft launch, and global launch. Use feedback from one phase to refine plans for the next phase. Support sales teams to identify necessary sales tools (such as an ROI calculator) and customer self-service deliverables (like a self-assessment worksheet).

**DISCOVER.** Conduct win-loss interviews and perform analysis for at least five accounts. Perform a retrospective on sales efforts and recommend revisions to sales tools. Perform a retrospective on revenue achievements and market share. Perform a retrospective on internal processes to identify areas to improve. Conduct opportunity scoring for the business model.

Tom has an idea, but he needs to ensure that his idea is attractive to a market full of customers. Tom needs to write down his ideas and plans so his vision is clear to both new

investors and new employees. Since Tom is “an idea guy,” he’ll want to recruit a product manager to join his team sooner rather than later. Tom isn’t really into the details; a product manager will ensure that everything is written down and kept current so Tom can focus on company issues instead of product issues.

Maybe you’re like Tom. How exciting! Not many people get to begin with a blank slate. You need to concentrate on the basics—personas and their problems—to convince your leaders, your investors, or your families to commit to your idea.

\* \* \*

Nowadays, creating products is less like building a home and more like maintaining one. After all, software, like a house, is never really finished. Who doesn’t have a list of home improvements on their “honey-do” list?

I’ve summarized how Meghan and Tom would implement the activities and artifacts using the QuartzOpen Framework as described in this book.

Meghan is a product manager. Tom is an entrepreneur.

In the following discussions, I’ll refer to a product leader or product manager—whoever is leading the product initiative—and how this role works with other teams such as executives, development, marketing, and sales teams. Later on, I’ll do a deep dive on specific roles and the challenges that are specific to product teams.

While my focus is primarily on software as a service, the methods described by the QuartzOpen Framework are relevant to any product or service.

Let's start your product adventure with *Discover*—the most important skill for product leaders.

## CHAPTER 1: DISCOVER

*"Find out what they like, And how they like  
it. And let 'em have it, Just that way!" —  
Fats Waller, Ain't Misbehavin"*

Meghan joined a company as product manager to develop and launch a new product. The president assured her the market was anxious for a solution. The president knew exactly what capabilities were needed. And regardless, the sales team was so great they could “sell anything to anyone.”

So Meghan ran with the idea. Unfortunately, during the initial planning, she started to hit resistance from the sales team. They wanted to give away the new product to close bigger deals; they said it needed additional features to make it attractive to new buyers; they needed her to call personally on each of their clients to explain the product.

She worked and worked and worked with very little result. Meghan decided to perform a retrospective on the business. (After all, it's never too late to LEARN.)

She challenged all the original assumptions. And (uh-oh) she learned that the product solved a problem that customers didn't have. Customers simply did not need or want the product. They certainly would not pay for it.

Many organizations complete a product and only then attempt to find customers who want it. That's totally backward! How can you know what features are necessary if you haven't talked to any customers? By customers, I mean both existing customers and non-customers—people who do not yet have a solution from your organization.

When you're managing a new product, question all assumptions. Don't expect that any decisions have been validated in the market. You'll want to approach your product with a healthy skepticism.

## Listen and Observe

Marketing gurus remind us constantly to “talk to your customers.” They're *almost* right. You should be *listening* to your customers. And many firms believe they are.

“Our salespeople and account managers listen to customers all the time.”

Except... salespeople don't listen to learn; they listen to *reply*. They listen for clues of what to say to close the deal.

“Our support people listen to customers daily.”

Except... support people don't listen to learn; they listen to *resolve*. They listen for clues of what to say to close the ticket.

One VP of Marketing told me proudly that he visited customers frequently. When I asked which customers, he said, “Our regional sales reps. After all, they’re our customers.” Alas, no; sales reps are delightful but they’re not your customers; they’re your distribution channel.

I don’t know why this is such a difficult concept for most entrepreneurs to grasp but, judging from the number of books and seminars on innovation and startups, *listening* and *learning* are rarely top-of-mind for most product initiatives.

Successful product leaders listen to learn; they listen to understand problems in their marketplace. After all, you don’t learn much while you’re talking.

Company executives generally believe that salespeople know their customers... and therefore the market. However, I’ve found that salespeople know *individual clients*, not necessarily the market full of clients. They know customer deals, not market opportunities. And salespeople have very short memories. They remember their last conversation but have forgotten the client feedback from a month ago.

Even more important than listening is this: you should *observe* your customers.

Do you have an idea for a product? Great! Go work side-by-side with people you hope will use it. Understand their processes. Check out their work areas. How do they get assignments and how do they deal with them? When do they write stuff down? When do they use a sticky note? How many stacks of paper are on their desks? When do they leave one program to do a process or calculation in another program?

As you interview and observe, you'll pick up subtle clues that inform your decisions. Your industry experience and market insights are valuable to understanding what you're seeing.

Product leaders need personal firsthand experience with the buyers and users of products—insights from future customers and existing customers.

Getting customer information secondhand doesn't work. It's like having sex: what you hear from others is nothing like your own personal experience.

In your role as a product leader, you must answer questions about the business, the market, the domain, and the technology. Your firsthand market experiences lend credibility to everything you say and write.

Product leaders listen to learn. They learn about customer workflows, attitudes, behaviors, challenges, frustrations. They also learn about their company's reputation, the effectiveness of their promotions and sales enablement, and the success and failures of their sales approach.



## Define Your Personas

*“When I decide just exactly who the song is a-going to help out, I can really scribble ‘er down in a hurry.”—Woody Guthrie, American folk singer*

The term ‘persona’ is used for the profiles of those who buy and use your products. And for many products, the person who buys it is not necessarily one who will use it.

Most teams have great success when focusing on the specific personas who represent targets for development and marketing. They define the behaviors, attitudes, and goals of the kinds of people we encounter with our products.

(Frankly, I’ve always had a little difficulty referring to customers as *users*. After all, only software developers and drug dealers refer to their customers as ‘users.’)

I’ve mostly stopped using markets and segments in favor of personas. Personas share a set of problems to be solved—or “jobs to be done” if you prefer—and don’t necessarily share a set of demographics or ethnographics. Yes, eventually you’ll need to use segmentation techniques for market sizing but begin with the problems to be solved for personas.

My first encounter with the term ‘persona’ was in Alan Cooper’s excellent book *The Inmates Are Running the Asylum* yet I’d been using the concept previously without this handy label. “Actor” is a similar concept used in UML, the Unified Modeling Language; “audience profile” has been used in the world of marketing for years.

Same idea; different labels.

When writing anything from a press release to a programming library, most folks write for themselves or their parents. And that's the value of personas. They provide a target for your writing. They remind you that you are not the target customer, nor are your families, your leaders, or your colleagues. Salespeople, marketers, and developers don't use your products; they may have good ideas but don't confuse them with customers.

When you first propose a product, you'll describe some basic information about one or more target personas—name, job title or role, maybe industry or geography. As you learn more about each persona, you'll update your persona documents with the latest information.

## **Meet Laura**

One of my favorite illustrations of personas is Laura, the university freshman. She's always had a mobile phone; she's always had a computer; she's always had access to the internet.

In almost every way, Laura is different from her college professors—who had none of these technologies when they attended university.

Professors who understand this disparity can relate their topic to Laura's school experience instead of their own. After all, the best professors connect with their students; the best marketers do too.

At a services firm, the new head of marketing insisted that all documents be rewritten for an 8th-grade reader (that is, a 14-year-old) since that's a common standard used for Federal

government consent forms in the US. However, the firm's customers were doctors who found such writing annoying and somewhat insulting. A clear persona definition revealed the disconnect between this marketing mandate and the target reader.

If you want to be heard, speak in the language of the listener—the persona.

The persona concept helps product teams focus on the specific problems of a type of customer. And the concept helps marketing and sales teams connect with them when they're ready to buy.

Smart product leaders are creating innovative products that address specific problems of their personas instead of just building a glob of features and hoping they will sell.

As for promotion, the old “perfuming the pig” approach of marketing has failed—no amount of perfume can overcome the stench of products that people don't want.

We create for product personas; we connect with market personas.

## **Personas Have Names**

Some companies use titles for their personas. Titles like “the decision maker” or “head of IT” or “HR rep.” One company used “managers” and “doers.” (WTH? Managers aren't doers? I guess managers don't actually “do” anything.)

But... shouldn't your personas have names?

I use “Robin” and “Meghan” for my product manager and product owner personas. I also use “Kevin, the world’s worst sales guy” and “Peter Principle, the CEO.”

Some teams use character names from TVs and movies, like “Data” and “Scotty” from *Star Trek* or “Dinesh” and “Jared” from HBO’s *Silicon Valley*.

These approaches work. However, some teams attempt to make the names generic enough to be gender-neutral (anyone remember “Androgynous Pat” from SNL back in the 90s?) but then find themselves writing “he or she” and “s/he” in their prose and the persona just starts getting ridiculous.

Try to make your characters realistic and recognizable. Your development and marketing teams will begin to care enough about your personas that they’ll create better products and promotions.

Names are important. Personas (and products) should have names.

After all, you named your kids, didn’t you?

## Anti-Personas

The value of personas is context; they provide an archetype for our teams to embrace. But you may also choose to build a few *anti-personas* for the kinds of buyers and users you want to avoid. After all, not every customer is a good customer.

Aren’t there customers you don’t want?

In 2013, Mike Anderson opened a sushi restaurant in Alexandria, Virginia, with a shocking new policy: no kids allowed.

*Because we strive to offer a cozy, intimate atmosphere, we ask that parents leave children under 18 at home. This allows us to provide the perfect environment for small groups and couples out on “date night.”*

Anderson’s target persona is an adult who wants dinner without the pandemonium of unruly children—their own or other people’s. The anti-persona is anyone with kids. “We thought parents just needed a place to give it a break, like an adult clubhouse,” said owner Mike Anderson.

It’s not that Anderson doesn’t like children; he owns other restaurants that cater to families. But by not allowing children, he can eliminate amenities associated with kids—high-chairs, happy meals, sippy cups, and so on—and create an atmosphere designed for those who want a break from youngsters.

“Couples have been loving it,” Anderson said. “We think we’re on the right track.”

Anti-personas are most helpful when evaluating features and capabilities—not every idea is a good idea for your product. If the idea is coming from an anti-persona, you can probably ignore it. I mean, ideally, you shouldn’t be selling to that kind of customer anyway.

## Personas Under Attack

But it seems the persona concept is under attack from some quarters. I've heard “personas don't work for us” and “we don't use personas; we write for ourselves.”

Invariably, these bad experiences come from “bad” personas—thin biographies of generic people with lots of nonsensical attributes. These unrealistic personas are laden with silly details about pets, children, cars, and other personal factoids. They're not based on personal experience with buyers and users of real products. They're out of sync with real people with real problems.

One marketing team created over forty personas based on job descriptions pulled from their personnel database. This initiative was an immediate failure.

Most products have three to five personas, not forty or even twenty. And a job description isn't the same as a persona definition.

Your goal is to communicate. Whether you're writing code or copy, writing doesn't have to be simple as long as it's clear to the reader. If your reader is a doctor, don't write for a 14-year-old. If your reader knows XML, you can write about classes and inheritance.

Know who you're writing for before you start creating either products or promotions.

## Build Your Persona Information

Personas provide insights on how to develop for and how to promote to your potential customers.

For consumer products, we often see demographics like age, education, and buying power. For B2B products, we may not care as much about personal demographics—such as family status, children, and pets—but even business buyers are still people. They have attitudes associated with their background, experience, and perhaps their family situation.

Consider the administrative persona for educational software. She is a divorced mom with two youngsters in day care. Does she want to work late to make some extra cash? (Answer: while she may need the extra money, she'll have to pay a late fee to the kids' caregiver.) But is any of this information relevant to how you develop or market your product?

A number of thought-leaders are questioning the value of common demographics. Must the parent of young children always be a mom and not a dad? Must a CEO be an old, white guy? No, those are *stereotypes*.

Personas should never be stereotypes; stereotypes are intended to demean. Personas are *archetypes*; they are meant to inform and even inspire.

My inclination is to focus more on their situations and problems, and avoid getting into sex, age, or number of kids and cats.

Common attributes for personas include titles or roles, industries, a typical day, and other products they're likely to

use. If you work with technical products as I do, you'll want information about their computing skills and their access to technology. What applications are used? What type of computer? What type of phone? What type of internet access? Do they prefer "open" platforms or the elegance of an integrated system?

So, the questions to answer in creating your personas are these: What information is relevant to your marketing and development decisions? Do you need demographic attributes or situational ones? Or both?

Consider this list of attributes recommended by Jennifer Doctor of **Harbor Light Partners**.

**Background.** Describe the persona's situation, physically and emotionally.

**Typical Day.** What do they do all day? What are they responsible for? How do they interact with co-workers and customers?

**Problems & Frustrations.** What challenges disrupt the persona day to day? How do they manage these?

**Motivations.** Why do they do what they do? How do they feel a sense of purpose and/or satisfaction?

**Influences.** Who do they listen to? For purchasing? For advice? When they need help?

**Goals.** What are they trying to accomplish? Why? How?

**Attitudes.** How do they feel about their job? Your company and product? The industry?

Fundamentally, we need to know how to find and market to our buyers and how to develop to our users.

More than one team has displayed posters of their personas in their work areas. I've seen them in team pods, conference



rooms, and public areas. The posters remind everyone who they are in business to serve. I've seen team members point to the posters and say, "Joseph will love this because..." and "This will really help Priya."

Personas bring your buyers and users to life.

Personas on based on real people. You simply cannot develop personas unless you have experience with real people.

## Size the Market

Since personas represent types of buyers and users within a market, they can also serve as a handy shortcut for sizing your opportunity.

How big is the product opportunity? To answer that, you'll need to know how many personas exist. How many instances of your personas can you estimate? There are many services that provide numbers, but you may not need to look further than LinkedIn. How many people on LinkedIn have your persona's title? That's a good start for market sizing.

For example, there are 6,293,900 results for product managers in the United States as of this writing. In order to determine the market opportunity for my workshops and coaching, I would do additional searches for product owner, product marketing manager, and other product management titles in the U.S. as well as other locations around the world.

You may need to get more precise later, but this rough market sizing is close enough to make a business decision at this point in your planning.

## Getting Market Insights

Where do you find personas with problems? Customer discovery.

Product leaders often ask me how to do market or customer discovery. They talk as if it's some sort of trick they can't seem to apply to their products. In any case, they usually can't answer the two most important questions: "Who are you building this for?" and "What problems are you solving for them?"

Many product leaders talk about technology and innovation and products and features, but few can discuss personas and their problems.

There's a belief in many tech firms that building technology is difficult but finding customers is easy.

The funny thing is that there are so many products begging to be re-invented. Look at the Nest thermostat and Nest Protect smoke detector. Thermostats and smoke detectors were thought to be commodities—functional and unattractive. Clearly a stagnant product category. Then Nest surprised everyone with a new line of products for the home that were both functional and beautiful.

Building technology is fairly easy; finding customers and their problems is the hard part.

## Problem Discovery: Interviews and Observation

*“What people say, what people do, and what they say they do are entirely different things.”—Margaret Mead, American cultural anthropologist*

Whether from colleagues or from customers, ideas usually begin with some form of a request. Someone has a problem and suggests that your product or company should solve it.

Often these ideas come in the form of feature descriptions, so you’ll need to work to get down to the root problem. Begin with a scenario and the people who are affected. Ask for an explanation of the basic problem in the language of your customer.

Ideally, we want to elicit a story about the problem from the customer’s point of view. This typically starts with the phrase: “Today I have to...” Probe for an illustration of the current scenario—at least one interpretation of the problem.

Next, try to get a description of the future scenario—a version of the solution. “Tomorrow I’d prefer to...” or “In the ideal world, I would...”

The proposed solution is likely to be somewhat narrow and usually feature-focused (which is okay) but a customer’s ability to envision possible solutions isn’t likely to be as strong as your product team’s.

It’s not the customer’s job to tell you how to solve their problem.

The customer's job is to explain the problem and the impact of the problem. What happens when this problem occurs? Is it annoyance? Lost customers? Reduced profits? It's helpful to know the difference when you start planning a solution.

You don't want to ask customers what features they need; you want to ask customers (and potential customers) what problems they experience.

Along the way, learn about the typical buyer or user of the product. What are their characteristics? What are their attitudes? What are their skills? These insights provide the basis for your personas. If you have developed a set of personas already, you'll be able to see whether the person asking for a feature aligns with your existing personas (which is a good thing) or represents a new type of customer (which is also a good thing).

You don't just want to *listen*; you also want to *observe*. People can rarely describe what they do in words. Watch their actions instead.

My friend Luke Hohmann of "**Innovation Games**" fame recommends a technique called "Me and My Shadow." Sit next to a person doing the job you're trying to improve. Yes, shadow them. Just sit and watch. With your mouth closed. You'll be amazed what you see.

This personal experience in the market from customer interviews and observation gives you powerful anecdotes and quotes that you'll use in your planning and promotion documents.

## Solution Validation: Experiments and Surveys

*“A survey cannot reveal what you didn’t know you didn’t know.”—Edward F. McQuarrie, *The Market Research Toolbox**

After you’ve found problems, you’ll want to validate potential solutions. Two techniques are helpful here: experiments and surveys.

For technology products, the most common form of experiment is the prototype. Ask your product team to put together a demo of a potential product or feature. This can be a mock-up, a video walk-through, or an interactive presentation showing how the product might support a specific workflow. Share the prototype with a small number of clients—three to five will usually give you the results you need.

Our goal is to get clients to tell us how a product based on the prototype will solve their problems. How would they use it? Who in their organization would benefit? How would they describe this to a colleague?

But beware: the fidelity of the prototype implies its completeness. (“It looks beautiful. When can I have it?”) In other words, when a prototype is extremely polished, executives, salespeople, and clients all assume the product is basically complete. It’s amazing how often salespeople sell the prototype, believing (or wanting to believe) that it is ready for the market.

Another popular form of experiment is the landing page. Create a simple page on the web describing your product idea

and see how many folks will opt to stay informed by sharing their email information. You quickly validate the solution by comparing the number of page hits to the number of people who sign up. And while you're at it, you build a list of potential buyers.

The survey is another technique for getting market feedback. Prepare a short list of questions to ensure your product assumptions are valid. Use a lot of closed question formats—yes or no, rate 1 to 5, ranking questions—to prove or disprove the anecdotal data you've already collected.

While surveys are great tools for getting data to support your hypotheses, nowadays they are so over-used that it is increasingly difficult to get broad participation. They seem to survey clients continually—after every hotel stay or airplane flight or online purchase. Our customers are getting “survey fatigue.”

You've probably also seen this: the word goes out that you're doing a short survey and other groups want to add “just one question” to the survey. And before you know it, you have dozens or hundreds of questions.

Successful surveys have only a few questions—they should take less than 5 minutes to complete.

Surveys by themselves rarely provide new insights. Instead, surveys provide the validation necessary to support the insights you learned from your interviews and observations.

## Look for Patterns

*“If we have data, let’s look at data. If all we have are opinions, let’s go with mine.”—Jim Barksdale, American executive, former president and CEO of Netscape Communications Corporation*

Any discussion of surveys invariably brings up the issue of statistical significance. For a sales guy, one customer anecdote is a completed research study. But for technical folks (particularly those who don’t like your idea), they want you to survey thousands. They don’t really want a survey; they want a *census* of every possible buyer.

You don’t need thousands. You can visit three to five clients for problem discovery and another three to five for solution validation. Use a prototype to elicit feedback on how the proposed solution will solve problems. If you need more data, you can survey a few dozen potential clients. That’s really all you need to reach a conclusion.

In your customer discovery and validation research, you’ll be looking for patterns. What do your customers have in common? What chores do they struggle with daily? How often do they spend time looking for something that was “right here just a moment ago?”

These patterns give life to your personas and provide the market insights that are key to building great products.

Use your own first-hand, personal market data to support product decisions. After all, every person in your organization

has an opinion. Strongly held opinions. And many of those opinions are in conflict.

Market data from your target personas is how you make rational decisions.

Every time you visit a customer is market research. If you write it down.

And learning doesn't stop here. You are always encountering new assumptions, new prototypes, and new messages to validate. Learning never ends.

## Opportunity Scoring

*"I know he can get the job, but can he do the job?" — Dan Hedaya as "Mr. Watori" in Joe Versus the Volcano*

So, you've got an idea for a product (or a service or a company). Most often, we rush to estimate costs and revenues. But first, you need to ask, "Is this idea a good fit for my organization?" Your resources are limited; don't waste them on unnecessary work.

Every organization has a long list of customer requests. Any could be a significant source of new revenues. Which should you pursue? Use ASPIRE+SOAR to determine which is the best fit.

Assuming you're solving a problem for a market, explore what qualifies you to deliver the solution and if you can make a business doing so. Whether you're just starting out or have an established brand, every team has capabilities to be leveraged



with new products. Yet it's often difficult to pinpoint your core capabilities, particularly when applied to a new product idea.

Should we, or shouldn't we? Find out using opportunity scoring.

I use the acronym "ASPIRE+SOAR" to help remember a set of questions for opportunity scoring. Each can be answered with a simple "yes or no." Notice we're not yet talking specifics about buyers or their problems, simply your ability as a business to develop and deliver a solution.

ASPIRE questions are focused on your capabilities:

- Do you have *authority* with customers to deliver this?
- Does this leverage your existing *systems*?
- Are you *passionate* about this idea?
- Will you offer an *innovative* way to solve the problem?
- Do you have *research* to support your claims?
- Does this idea leverage your existing *expertise*?

SOAR questions are focused on your goals:

- Does this align with your *strategic* vision?
- Does this supplement or add to your core *offerings*?
- Will this improve your competitive *advantage*?
- Does this offer significant financial *returns*?

These business factors help you determine how much you can leverage corporate assets to develop and deliver a new product idea.

## **ASPIRE: Could You?**

One firm acquired some technology and wanted to turn it into a product. Unfortunately, they ran into difficulties at every step.

Why? They didn't have systems or passion; they bought another company's innovation, but they lacked the expertise to leverage it. A reevaluation using ASPIRE+SOAR revealed the disconnect.

Another firm had a list of twenty items, far more than they had resources to deliver; they could only run with a few of them. An ASPIRE+SOAR assessment helped them determine which to pursue and which weren't a good fit.

### **Authority**

Do you have authority with the market to deliver this? Do your current and potential clients expect you to solve this type of problem or offer this type of solution?

Sure, you can develop almost any product or service but will your target market embrace it? Does it align with your brand? If your company has a reputation in a domain or market, you're likely to have the authority to offer additional products to your existing customers.

Who is your favorite vendor? Why do they have credibility with you? They have earned the authority to offer you products in a specific category. Your favorite software vendor can easily offer new types of software but you'll probably be skeptical if they try to offer hardware or cars or sports equipment.

I love my Nest thermostat. My positive experience with their thermostat granted Nest 'authority' to sell me additional household products. When Nest started offering smoke detectors, I was willing to pay a premium for their new product.

Likewise, Apple has been delivering high-quality, elegantly designed products for years. As a result of their authority,

many clients (including me) bought an iPad before they'd actually seen one. I don't trust many companies enough to buy a new product so blindly; do you?

Salespeople have known this for years: it's easier to sell more products to satisfied customers than it is to find new customers for existing products.

Should Nest or Apple start selling sports apparel? Or food delivery? They may have the capability but, no, I don't think their authority reaches that far away from their core business.

Do your customers expect you to deliver this? What's your authority and with whom?

### **Systems**

Does this idea leverage your existing systems? Are you ready to support your new product initiative with proven business and technology systems and methods?

For example, a company offering curriculum for teachers wants to offer educational courses directly to the public in the form of video presentations. They'll need to acquire the systems to produce and deliver video content. They'll have to figure out how to take money over the web instead of using purchase orders as they do today. They'll need a way of protecting content so it's not shared or pirated. And they'll have to evaluate whether their existing content is effective when delivered via video instead of the classroom setting they already know.

Many startups encounter this lack of systems as they try to scale their business. They've put all their focus on product but not on the supporting business systems.

You'll either need to leverage existing systems or somehow acquire them (via partnerships or "acqui-hiring"). Do you have the necessary systems in place to offer this product or solution?

### ***Passion***

Are you passionate about this idea? Or as Simon Sinek says, "What's your 'why'?"

One of the things you notice about successful companies is their passion, their zeal. Passion for the domain. Passion for customers and passion for solving their problems. Their developers are thirsty for information about the people who use the product. Their marketers always want to know more about how customers buy and which messages resonate with them. Good salespeople sincerely want to help customers solve their problems.

Some say "a good manager can manage anything" but that hasn't been my experience. Instead I look for managers who are passionate and knowledgeable about their domain, their products, and their customers.

Passion seems to be a key element of success for most organizations. Passion for customers and their problems is what makes employees go the extra mile. Furthermore, passion is what keeps you going when you're faced with difficulties.

Seasoned executives have learned that passion for customers translates into value for shareholders.

What's your passion? Will your organization be excited about this idea next year and the year after?

## ***Innovation***

Will you offer an innovative way to solve the problem? Have you invented something new? Do you have a unique approach?

An innovation could be a new type of technology or a new delivery model, a new way of pricing or packaging; you could innovate by turning a product into a service or a service into a product. In short, innovation is a new way of solving a problem.

Consider the Internet of Things—one of the “hot” topics of 2020. It’s so much more than smart light bulbs and phone apps. I want my freezer to tell me when it is having trouble maintaining temperatures and when my food is in danger of spoiling. I want my heater to tell me when it is time to change its filter. I want the basement to tell me if the floor is wet, meaning my water heater has failed. Or better yet, the water heater should alert me before it fails. It will be interesting to see how vendors solve these issues with new technology innovations and inventions.

In fact, most innovations solve a problem that is already solved; it just isn’t solved as well as it could be.

You can also be innovative with pricing and packaging. Years ago, Microsoft packaged Word, Excel, and PowerPoint into the Office bundle which solved a purchasing problem for their industrial buyers. Nowadays they offer many such bundles, each targeting a specific persona.

Many products today are moving from one-time licenses to subscriptions. Adobe was one of the first to adopt this approach. In 2013, Adobe moved from product pricing to a

cloud-based subscription model. Many customers were furious with the idea of renting software. Yet Adobe's new approach to packaging made it easier for people to buy, resulting in \$6 billion in annual revenue in 2018.

What's your innovation or invention? What's your intellectual property? Do you have a unique way of solving the problem?

### **Research**

Do you have research to support your claims? Other than yourself, who agrees with your approach? Perhaps domain experts and scientists have published studies. Or you may have customer success profiles that support your fundamental conjecture. Research helps you prove it.

Claims that are unprovable or unsupported erode your credibility with both customers and colleagues.

A company is building an application to help guide and coach romantic couples. The app gives advice on activities likely to please your partner. Don't you wonder: How do you know you're getting good advice? What does the vendor know about counseling couples? What is the source of their recommendations? In this case, the product team can point to specific research that shows which actions are most effective at building and maintaining committed relationships.

Research is especially helpful when going against commonly held beliefs or accepted standards.

Most sales leaders contend that "relationship building" is the most important skill in selling but authors Matthew Dixon and Brent Adamson learned otherwise. Their book, *The Challenger Sale*, describes a selling approach in which the

salesperson actively teaches their prospect about trends in the industry. Based on a study of thousands of sales teams across multiple industries, the authors revealed this approach delivers higher sales performance consistently.

It's hard to argue with research.

Do you have research to support your claims? Research provides independent proof of a successful outcome.

### **Expertise**

Does this idea leverage your existing expertise? How widespread is the experience of your team on this type of product in this type of market? Is the expertise found in only one or two employees or does everyone on the team (or everyone in the organization) have the necessary skills?

I worked as an advisor to a company and many of their struggles were related to the absence of expertise. The sales team couldn't recognize a good client from a bad one. The developers were building features they thought customers might want, not features clients would actually use. Both turned to the product manager who didn't really understand the problems in the market or the common methods in the domain. The result was a hodgepodge of partially implemented capabilities driven by a diverse set of customers. The company failed because of an absence of expertise.

Do you have the necessary expertise to succeed with your product idea?

## SOAR: Should You?

You and your organization may have the *ability* to create a product offering but should you? Can you do so profitably?

ASPIRE represents a set of questions for evaluating multiple opportunities. But should you bother? Will the idea fit your organization's directions and portfolio? These are the next set of questions using SOAR.

As firms look at the many, many things they could do, the firms which survive are those focused on the vital few—the ones where ASPIRE and SOAR meet.

Consider these questions for alignment with your organizational goals.

### **Strategic**

Does this idea align with your strategic vision?

Strategic vision links the present to the future, showing how you intend to move the organization to its next level of performance. Much has been written about strategic vision and alignment. Context vs core. Must-have versus nice-to-have. The question here is whether the idea is a key component of your long-term strategic vision. That, without it, your company's long-term viability is in jeopardy.

For example, a firm could decide to expand its portfolio of offerings to better support an existing market or decide to promote its existing set of offerings to a new market segment. The choice between new product initiative or new market focus is a strategic decision.

In the early days of Amazon, the company focused on selling books on the internet. After all, books were easy to sell



and easy to ship. Over time, more product categories were added, such as CDs and DVDs, with the same “easy to sell and ship” characteristics. Jump forward a decade and look at Amazon’s introduction of Kindle readers and Fire tablets. Not quite as easy to sell and ship, and much harder to develop and promote than previous products. Despite these drawbacks, offering readers and tablets was a strategic decision—the devices were a gateway, moving Amazon and its customers from physical goods to electronic goods. Want a new book or some new music? Just download it.

Strategic vision provides focus for business, market, and product portfolio decisions.

### *Offerings*

Does this product idea integrate with your core offerings?

Business book authors have learned that books sell more speaking engagements and speaking engagements sell more books. Rock bands have learned that T-shirts and vinyl albums are souvenirs that dramatically increase their revenues at concerts. For Amazon, offering readers and tablets meant they could offer books, music, and video electronically, a nice fit with their existing portfolios.

It makes sense for Apple to offer monitors, keyboards, and track pads as complementary options to their computer business. But should Apple (or Nike or Comcast) get into the house-building business? Probably not; these are not aligned with their portfolio of offerings.

Salespeople love having additional products to sell to existing customers but are confused when new products don’t complement the existing product suite.

New offerings should fill a gap in the market and complement your suite of products. Product offerings should reveal a coherent portfolio strategy, centered around markets and customer needs.

### ***Advantage***

Will this improve your competitive advantage over alternatives?

Superior products and services are compelling reasons to buy your product suite over a competitor's. For technology products, an installation service or an extensive do-it-yourself online video series can provide competitive differentiation. Your pricing and packaging decisions can be competitive advantages so look at flexible pricing and product bundles too.

Just because your competitor has a feature or a product option doesn't necessarily mean that you should offer it too but it does make sense to evaluate whether the idea should be pursued.

Some projects help you jump ahead of your competitors; others may be necessary just to catch up to them. Either way, new offerings should improve your competitive advantage.

### ***Returns***

Does this offer significant financial returns in new revenue or cost savings?

A new product or suite of products focused on the problems of a well-understood market segment is likely to generate healthy revenues for your organization in the mid- to long-term. A troubleshooting wizard or well-crafted mobile app will probably not generate new revenue but may reduce customer support calls and costs significantly.

Some projects can easily be justified when considering new business revenue while others offer value by reducing internal costs associated with manufacturing, maintenance, support, and selling. Without completing a full financial forecast, you can still determine a rough scoring of revenue growth or cost savings within some reasonable timeframe, typically within three years.

While not the only factor to consider, products are ultimately judged on their contribution to the bottom line through increased revenue or cost savings.

## Leverage Your Capabilities

The likelihood of product success dramatically increases when you can leverage existing capabilities and competencies.

Compute a score for each opportunity and weigh them against each other. After all, you have limited resources—in development as well as marketing and sales—so you need to ensure you’re working on the top priorities.

Use ASPIRE+SOAR to identify product ideas with the best potential for your organization and to compare multiple potential offerings.

Answer the following questions “yes” or “no.”

☐ Do we have authority with customers to deliver this?

☐ Does this leverage existing systems?

☐ Are we passionate about this idea?

☐ Will we offer an innovative way to solve the problem?

- ☐ Do we have research to support our claims?
- ☐ Does this idea leverage our existing expertise?
- ☐ Does this align with our strategic vision?
- ☐ Does this supplement or add to our core offerings?
- ☐ Will this improve our competitive advantage?
- ☐ Does this offer significant financial returns?

A “Yes” is worth one point, a “No” is worth zero. Add up the total of “Yes” responses to determine the final score.

Obviously a 10 out of 10 is the best. You should consider “fast tracking” an item with a score of 9 or 10, and seriously explore any items with a score of 7 or 8. Those items with a score of 6 or less should be put on hold and perhaps reexamined in the future.

I often get calls from people asking advice about their plans of starting up a business or creating a product. They ask, “Do you think it’s a good idea?” and “Do you think I should pursue this?” Golly, how would I know?

I challenge them to examine their capabilities and their expected results.

A friend wants to offer a new service. I think he can probably *do* the work but can he *get* the work? Alas, he doesn’t have any original, innovative tools. And he doesn’t have much of a reputation in this area. Yes, he has some expertise but is that enough? Will anyone believe he’s the one to call?

Opportunity scoring is particularly important when you have multiple product ideas and limited resources (which, of course, is every organization). What idea should you pursue?

Lead your team in a quick opportunity scoring exercise using ASPIRE+SOAR.

Ideally, you'll maintain a long list of product and service ideas, each with an opportunity score. Whenever you have the resources to take on a new opportunity, you can pull the next highest priority item from your queue.

Once you've narrowed the field of ideas from the possible many to the realistic few, you're ready to evaluate the idea fully.



## CHAPTER 2: COMMIT

*“The hard part is implementing the decision, not making it.”—Guy Kawasaki, author, entrepreneur, and venture capitalist*

Executives get ideas based on a conversation or a flicker of an idea while in the shower. During staff meetings, they chat about what could be and “wouldn’t it be great?” They produce a few clever phrases, perhaps a few napkin drawings. They get really excited about possibilities.

And then what?

One sentence from an executive typically results in a huge amount of work for a product team. A cocktail napkin idea may take months or even years to define and deliver.

But the one who ultimately “owns” an idea is the one who writes it down and keeps it current.

A product leader—in the software world, it’s often a product manager or product marketing manager—who converts the leadership team’s strategic vision into tactical action—

vetting the idea with the market, justifying with a business case, and translating the big idea into actionable artifacts, particularly stories, capabilities, and roadmaps.

A product leader turns the big idea into all the persnickety little details that make it something real. The product manager is the one who writes it down and keeps it current.

A common problem in many organizations is the ill-conceived desire to have perfect information before it can realistically be known. It's simply too early for revenue and cost estimates; there are too many unknowns. At this point, we're not looking for funding; we don't know the specifics of the features or the price point or the marketing plans. We're just trying to get approval to further investigate the product idea.

If it's a bad idea, let's stop immediately; if it seems a good idea, let's fast-track it.

## **Define Your Product**

Before you get into the details—personas and stories and plans and prices—take a moment to define the product idea at a high level and gain some level of approval for the idea to go forward.

No matter where you are in the lifecycle of your product, you'll need to define what “it” is... with a name, a basic message, short and long descriptions, and so on.

Perhaps the easiest way to share your ideas with others quickly is to give it a name and compare it to another product: “it's like \_\_ for \_\_.” It's like [something they already know]



for [new market or application]. iTunes, Netflix, and Airbnb are all commonly used as reference products. I use an ebook manager called **Calibre**; it's like iTunes for ebooks. Spotify is like Netflix for music. DogVacay is like Airbnb for dogs.

This reference product could be the basis for a good product code name; you could use “iTunes for ebooks” as a code name until a branding team has landed on a well-researched name.

A code or project name is a helpful placeholder for your new business or product idea. For instance, “Newco” is a common code name for a new company initiative and it serves its purpose as a simple placeholder. “Tom’s product” is a handy name too. It’s just a temporary shorthand reference.

Beware: popular project names tend to stick so don’t pick a name that people actually like. You don’t want the name showing up in code or in external roadmaps. Bland names are good; slightly disgusting names work too. “I’m working on ‘Green.’ Tom is almost finished with the Earwax project.” Clearly these bad names won’t leave the building.

The key is to pick a product code name that must be abandoned. You’ll pick a good name later, working with your marketing and branding folks using information from existing and potential clients. You’ll get feedback from salespeople and the general market. Since you haven’t yet done market validation yet, it’s too early to pick a “real” name.

## Identify the Competitors

As we saw in the last chapter, opportunity scoring helps you determine if your idea is a good fit. Now you have to prove it.

The first step is defining your competitive landscape. Who are your competitors? You may already know them but here's a tip: Do a search for each of your major features and see what competitive products come up. Some competitors will be familiar to you; others may be new. There will be some who are not a threat, but you'll want to profile them anyway; one of your sales guys will surely stumble across them one day and freak out.

For each competitor, you'll need to describe their product and positioning. You'll find most of what you need to document their pricing and features by scanning their web site and reviewing other published documents.

## Calculating Competitor Threat Levels

Here are five questions to calculate a competitor's threat level quickly.

- Are we losing deals to this competitor?
- Are we likely to lose future deals to this competitor?
- Does the competitor have brand awareness in our market?
- Is the competitor investing in promotions in our market?
- Does this competitor have more product funding?

Answer each question with a yes or no. If you answered yes to all five questions, you'll need to do a complete profile of this

competitor. If only one or two are true, you'll want to explain to your sales teams why this competitor isn't a threat.

Not all competitors are worth worrying about. You're likely to find several products in your category that don't compete on features or target a different set of personas. You'll want to score the threat level of each.

## **Pricing and Revenue Forecasting**

You probably have some ideas for your pricing options—maybe a plan for 1 to 10 users and another for 11 more. Or Silver, Gold, and Platinum packages. Or maybe you're doing licenses, so it's just a one-time purchase (like movie tickets and music concerts).

While gathering information about your competitors, remember to capture their pricing options. You'll want to compare your proposed pricing to theirs later.

Many product teams approach pricing from a competitive standpoint: “How little must we charge below the competitors’ prices?” Or they approach pricing from a cost standpoint: “How much can we charge to cover the costs of development, marketing, sales, and delivery?”

Instead of cost-plus or competitor-minus pricing models, pricing experts agree: your buyers don't care about your cost; buyers care about their value.

Consider external hard drives. You can backup your computer to pretty much any USB drive—they're mostly commodities. But what about a drive specifically designed for backing up pictures? After all, losing all your photos of family

and friends would be devastating. Wouldn't the typical mom or dad pay more for a way to ensure their pictures are safe?

Understanding value means understanding the impact of the persona's problem. How much they lose in revenue or productivity or convenience gives you an indication of the level of pain and the value of solving it. In your customer interviews, you'll want to ask questions like "What would happen if this problem occurred?" and "Who would be impacted?" and "How much would that cost you?"

A simple way to look at pricing is to use a 3 by 3 matrix. The vertical axis is the perceived value associated with solving the problem; the horizontal represents your competitive differentiation.



The closer you get to the lower-left, the more commoditized your offering. If it's a simple problem and there are lots of available alternatives, you won't be able to charge very much. With a higher concentration of competitors, you'll need a clear differentiator to cut through the noise of a crowded marketplace.

The closer you get to the upper-right, the more you can charge. That is, if you're addressing a significant problem and have a differentiated solution, you can charge much more. That's where your ASPIRE attributes (particularly *innovation* and *expertise*) translate to higher value for your solution compared to competitive alternatives.

Now this is where it gets interesting.

The price reflects competitive positioning.

How much you charge relative to your competitors is predicated on how your company and product are perceived by buyers. Are you a premium company and product? Or are you a commodity player? Are you upper-right or lower-left in the pricing diagram?

If you're the leader, you generally charge a premium. After all, you have the brand, the reputation. And it confuses the marketplace when a premium company offers a commodity product.

If you're not the leader, you might choose to price equal to or a little less than the leader in your category. In effect you're saying, "We're as good as they are but cheaper."

Hey, it's okay to be a commodity. But it's difficult to charge a premium price when you're a commodity player. You may be able to charge a premium for your unique capabilities, but you'll need to use commodity prices for "me too" products.

And just because you can charge more doesn't mean you have to.

Take a moment to reflect on your product goals. Are you going for market share or revenue? If you have a market share

objective—that is, you want to get the product into as many hands as possible—set the price lower. If instead you want to generate as much revenue as possible, you should set the price higher.

Your costs and revenue plans are irrelevant to your customers but, of course, they're quite relevant to your business case—if your persona's perceived value is less than the price you can charge, you have a flaw in your business model, not a flaw in your pricing. Don't forget to take into account your company's position in the competitive landscape. And set your prices accordingly.

Pricing. It's a little bit of magic. But it's critical to your product success.

## **Allocate a Product-Specific Development Budget**

Despite the adoption of agile philosophies, many product executives still demand detailed time estimates and costing at the feature level. As someone who has worked with product teams for decades, I can tell you time and cost estimates of features just aren't reliable.

Or, as we say in the consulting biz, “How's that been working for you?”

Here's the reality. If you insist on a date, the team can deliver whatever is done by that date. If you insist on a set of features, the team can deliver them on whatever date they're done.

What doesn't work is mandating both what to deliver and when to deliver it. That results in low product quality and team burn-out.

Execs and product managers ask, "How long will this take?" and their team correctly replies, "We'll have to fully specify the solution in order to provide an estimate." Which is true—they honestly can't tell how long it will take until they know what "it" actually is.

One of the key concepts of Lean manufacturing is to avoid waste. It's downright silly to waste time developing a design specification just to get a cost estimate. And honestly, preliminary requirements often lack the detail necessary to design a solution.

I mean, really, how accurate have your feature estimates ever been? Asking developers to provide scope and time estimates on individual items is like asking salespeople to forecast specific deals and contract values years in advance.

What to do instead? Prioritize against a development budget.

Instead of thinking in terms of features and schedules, budget the teams you can afford and have them deliver as much as they can. After all, software is never finished. There are always new bugs to fix, new technology to support, and new features to add.

Start by allocating development dollars by product. Product A gets a million; product B gets two. Have each team prioritize their backlogs for business value and always work from the top of the queue.

In essence, this approach tells you to spend money on development aligned with the importance of the product within the portfolio. At the most basic level, your product development budget specifies the number of developers dedicated to your product. With this allocation, team staffing effectively articulates your portfolio strategy.

Build new functionality against the product budget. Stop building when you run out of time or money.

## **Allocate a Product-Specific Promotion Budget**

How important is each product to your revenue objective? Use that answer to guide your promotions budget. Just as you did with the product's development budget, allocate a percentage of the overall promotion budget to each product.

I think of it this way: this year's development budget drives *next year's products* while this year's promotion budget drives *this year's revenue*.

One marketing team was dealing with five different products in two major markets. Over time, promotional spending had evolved into a "first come, first served" situation. The product managers learned to get their programs set up early in the year to ensure they got some promotion money for their products. We recommended a change from the "big bag of money" budget to micro-budgets for each product. Each product had a specific amount of money for promotions throughout the year.



Budgets aligned with revenue goals generated vastly better results.

The key was to get executive buy-in for what percentage was allocated for each of the different products. The newer products got a larger amount while the older products got relatively little. Profits from the older products funded the promotion of the newer ones.

## **Build a Business Justification**

Often it seems that writing a business plan is just a waste of time. Ash Maurya, author of *Running Lean: Iterate from Plan A to a Plan That Works*, jokes that a business plan is “a document investors make you write that they don't read.”

However, the value of a business plan or one-page canvas is as much for the writer as for the reader. Creating a plan forces you to consider all aspects of your product opportunity—the personas and problems, positioning, competitors, revenue projections, and cost assumptions.

There are numerous approaches to building a business justification for a product. Some organizations use an extensive text document; others use a presentation. Many today use a one-page business model canvas.

Regardless of the format, every business justification should include the following:

- Product concept
- Problems to be solved
- Target personas and market size
- Competitive landscape

- Sales approach
- Pricing models and revenue forecast
- Development budget
- Promotion budget

At the end of your business presentation, you'll be seeking an action. That's "The Ask." What are you asking your executives to authorize?

You'll use this information next.

## **Gain Executive Commitment**

Want to know if your executives truly support a product initiative? Get their authorization to hire someone.

With your business justification in hand, you're now prepared to brief your executive team if you have one (it could be your spouse). You have everything you need in the artifacts you've produced so far. Set up a meeting to walk through them.

Your presentation should include three types of information: stories, data, and process. Some executives make decisions based primarily on customer anecdotes and stories, so be prepared to quote customers by name. You'll want to share conversations you've had with potential customers as well as the feedback you've gotten from internal teams, especially salespeople.

Technical executives tend to rely on data—the results of your surveys will provide credibility here.

And many executives are process-oriented so they will need to see how you arrived at your conclusions—the process you followed and the formulas you used.

If you have a one-page business case, use it as an overview supported by slides for each of its elements. Save your slides for the concepts and the bottom-line results. Items with lots of text—such as your pricing model and revenue forecast—should be printed as documents. Imagine a simple chart with annual revenue projections supported by a printed version with the individual line items. Don't annoy your executives with slides they cannot read.

Your final slide or discussion is “the ask.” What are you asking the executive to authorize based on this discussion?

Here's the ask: It's time to take the idea the rest of the way. Will you authorize a new headcount?

No, you cannot use existing resources; they're already over-allocated as it is.

No, you cannot outsource it; institutional knowledge needs to stay in the company.

We need to hire a full-time product leader.

A full-time person ensures the company is completely behind the idea—if they're willing to hire someone to lead it, they are truly committed to it. And if they can't (or won't), then the company clearly doesn't have the resources to move the idea to the next stage.

My rule is this: Any idea worth pursuing requires a full-time manager.

Your leadership reveals their support for a product justification when they authorize a new employee to manage it.

You've evaluated the idea and your leaders (or investors) have seen its merits. There's still a lot to do to turn an idea into a product. The next step is to start describing, building, and delivering it. This is where the fun begins.

## CHAPTER 3: DESCRIBE

*“Learning everything about the customers becomes the most important focus of creating the product strategy. When companies talk about the problems of the customers, they are saying, ‘Look at what you’re dealing with. Look at how we want to help.’”—Jared Spool, founder, User Interface Engineering*

Over the years, product leaders have attempted many different approaches for describing what customers need. We’ve used traditional requirements and use-cases; we’ve tried open discussions; we often use some form of “story” today.

Stories are illustrations of typical scenarios using your product. A story should be big enough to generate interest when it appears on a roadmap.

The difficulty is that many stories today aren't really stories; they are more often short descriptions of features or tasks. And let's face it: most user stories are boring.

The formulaic approach doesn't result in a good story. They follow a standard format like, *As a [role], I want to [action] so I can [outcome]*. Read more than a few of these and your eyes glaze over.

Let's change that.

Stories should be... *stories*.

The Product Story technique creates a narrative that piques your interest. A Product Story addresses a market problem with a complete scenario for a persona. It describes a capability with business rules: needs, constraints, and outcomes. It's likely that one Product Story will address multiple problems for one or more personas.

## Persona Problems are Product Stories

*“When people tell you something’s wrong or doesn’t work for them, they are almost always right. When they tell you exactly what they think is wrong and how to fix it, they are almost always wrong.”—Neil Gaiman, English author of short fiction*

In your market observations and interviews, you learned your product's personas and discovered problems to solve.

Bring a group of smart designers and developers together to talk about personas and their problems. Tell the team about the

business opportunity found in solving those problems and participate in a brain-storming session on how you might solve them with your expertise and innovations.

Product leaders share the problem and participate in brainstorming possible solutions.

This type of meeting is called a discovery session or sometimes design thinking. At its core are stories. You know, actual narratives. Like, “I talked to Jennifer who is a divorced mother. She works full-time so she puts her daughter in day care. Let me tell you what keeps her up at night, what stresses her out.”

Let’s learn more about Jennifer and her problem.

*Jennifer is a divorced working mom and she worries about her daughter. Sally is a young school-age child, and, after all, what parent doesn’t worry these days? And to make things worse, Jennifer’s ex-husband is pretty unreliable; he frequently makes commitments to pick up Sally and then forgets completely.*

*For example, one afternoon at 3:30, Jennifer learned that Sally is still at school when she was supposed to be at soccer practice. Jennifer wondered “Why did I ever marry that jerk?” as she raced from the office to pick up Sally.*

*Jennifer would feel a lot better if she always knew where Sally was—that is, when she was where she’s supposed to be, and more important when she’s NOT where she’s supposed to be.*

## Same Story, Different Persona

You're likely to find that your product idea appeals to multiple personas. If so, another Product Story explains the same scenario for a different persona.

*Frank worries about his father, Bob, because he lives alone and is starting to show signs of dementia.*

*Periodically throughout the day and evening, Frank wants to verify that his father is where he's supposed to be and specifically wants to be know when his father is not at home.*

From these examples, we see two personas with pretty much the same problem: ensuring family members are where they're supposed to be.

In our team discussions, we may choose to combine them into a single persona who is concerned about a family member, regardless of age. However, some aspects of their product usage are sufficiently different that you might keep them separate; a child and a senior adult will likely need different monitoring approaches.

## Define Features and Capabilities

Once you have a problem defined initially (think of it as a draft), you'll want to have a conversation with your development lead or product team—what I call a discovery meeting—to uncover the detailed stories.

Ron Jeffries, one of the creators of Extreme Programming (XP), advocates the form of “**Card, Conversation, Confirmation**” for stories. The “*Card*” is merely a placeholder



for the story; each story requires a discussion with the product team and a method for determining when the story capabilities have been delivered successfully. Unfortunately, many teams seem to have forgotten the *Conversation* and *Confirmation* parts—the card by itself isn’t very valuable.

The discovery meeting involves both a product manager and the product team who discuss only a few problems or stories at a time. This meeting reveals ambiguity as well as unarticulated needs, constraints, and outcomes. Some of these will be in your head but not documented; others will be replies to inquiries. “Should Jennifer’s husband be able to revise Sally’s calendar?” might lead to a long discussion about roles and permissions—which will likely lead to more stories.

That said, imagine the healthy interaction in the product team as they come together to design a solution to these problems. The ideas come at a fast and furious pace:

*We’ll need some type of tracking device, like a bracelet or necklace. I’m assuming it’ll be GPS-based.*

*What if those don’t work? Maybe we should look into embedded chips—maybe put them in their shoes.*

*Would it be possible to put a tracking chip under the skin, or is that too creepy? And that’ll probably mean we need FDA approval or something. Let’s put that aside for now.*

*Does anyone know anything about “geo-fencing?”*

*There are a couple of ways we could alert the customer: we could make an automated phone call or an SMS alert.*

*I’m thinking we should do an app; then we can do setup and alerts using the same platform. We should definitely do Android and iPhone. Let’s get somebody to find out if there are other phones to worry about.*

*I read recently that Apple and Android have at least 90 percent of the market, so I think we're okay with just two apps.*

*Personally, I sleep pretty soundly. I think we ought to have a fairly loud alert when people aren't where they're supposed to be.*

*Well, I'd let the persona make that decision. Or at least, make the sound a user-specified value including vibrate.*

*Hey, I'm getting confused about who is who here. Can we focus on Jennifer and Sally and just use their names from now on?*

*Good idea. But let's also remember Frank and Bob because they have scenarios that are a little different. For instance, Sally may be more inclined to wear a bracelet if her mom insists. But Frank's father might be resistant—we may have to trick him into wearing it somehow.*

*Hey, my grandfather wears a copper bracelet—he's convinced it eases his arthritis. I think it's silly, but he doesn't. Maybe we can make a copper version of the bracelet.*

*And so on....*

Ultimately the team decides to build a setup and alerting service plus a device to be monitored.

*(Personas) Jennifer and Frank have family members who need to be monitored.*

*(Outcome) Know where family members are and where they're supposed to be.*

*(Story) monitor location of family member(s).*

*(Story) report current location with scheduled alerts.*

*(Story) alert on departure from geographic area (home, school, work, etc.) based on scheduled times (i.e., "geofencing").*

Each of these key stories will need to be expanded to include needs, constraints, and outcomes. The outcome statement may be the most important—because it really doesn't matter if you meet the individual needs and constraints if you don't ultimately achieve the outcome.

The needs, constraints, and outcomes will be the measure of whether a story has been achieved or not. It's not "done" until all the needs, constraints, and outcomes have been addressed.

The outcome statement is also potentially a positioning message. It's what you'll tell your agencies and internal folks to use in any promotions. Your creative folks may do some wordsmithing to help it resonate with buyers.

*"WhereRU—Know where your family members are, especially when they're not where they're supposed to be."*

In terms of market validation, before creating any new technology, the team could create a web page inviting people to pre-register for the service and see how many people sign up.

The team could test the product concept by building the setup and alerting service. They could initially perform the setup and alerting using a completely manual system: clients set up alerts via a phone call to the support team that maintains the alert information in a shared spreadsheet. Based on the alert information, a support person would simply call or text the client, depending on what was specified in the setup.

In other words, without any programming at all, they could provide this service to a small number of customers.

If all goes well, the team would automate more and more of their manual processes: moving from a phone setup to an app, moving from a shared spreadsheet to a database, and so on.

Don't get trapped into a conventional approach to writing requirements or formulaic user stories. Tell a series of actual stories featuring real-life personas and their real-world problems.

## Describe Needs, Constraints, and Outcomes

For another example, let's review the story details for a capability for a multi-user admin system.

*STORY: Update roles, assignments, and permissions*

*Product administrators need flexibility to control what their people can view and edit. Each user will be assigned one of four roles: Admin, Editor, Author, or Reader.*

The needs, constraints, and outcomes for the story related to Author permissions are the business rules defining how the system will behave, including:

### **Outcomes**

- Administrators determine what users can access to prevent interference in areas beyond their job function.

### **Needs**

- Authors can read anything.
- Authors can create, edit, and delete artifacts of types they are granted regardless of who owns the artifact. For example, if Joe is granted permission to author Products, both Joe and Tim can edit Products managed by Tim.
- Authors can edit and delete artifacts they are assigned even if they do not have privileges for that artifact type.

- Authors assigned to Launch Plans as Marketer can only create and edit DETAILS. They can delete what they add but cannot verify or unverify.
- Authors assigned to Product Stories as Developer can only create and edit DETAILS. They can delete what they add but cannot verify or unverify.
- Authors with privileges on Product artifacts also have privileges on Product Stories, Launch Plans, and Requests, even if those privileges are not explicitly granted.

### **Constraints**

- All users must be identified in the user table for the account in order to access Playbook.
- A user must be specified as an Admin, Editor, Author, or Reader. Only one type can be chosen.
- Authors assigned to a Product cannot change the product assignment.
- Authors cannot use tools that impact more than one artifact (Quick Prioritization, Competitive Matrix) unless they have been granted authority for the associated artifact types.
- Authors can only use tools that are specific to one artifact if they own the artifact or are granted access to the artifact type.
- Authors can create comments but can only edit and delete comments they create.

For this project, we paid particular attention to consistency. If we did one approach for the product manager and developer, we generally duplicated the approach for the product manager and marketer.

The team could have waited to get into the implementation before asking for clarifications and explanations. In this case, we hashed out the details in advance in multi-hour meetings when discussing how we needed the system to behave.

You may have several discovery sessions as you learn more about each issue and the product team starts working on its implementation. Better to discuss each item as a team when you're all really engaged than to have to respond when you're distracted by something else.

Yes, it's tedious. But better to talk it through now than wait until it blocks the team from completing the story.

And notice, these details don't tell the product team *how* to do it; they describe *what* needs doing.

## The Problem of Story Detail

The level of precision in stories is an industry-wide problem. In general, product managers want to convey high-level scenarios (or problems) while many developers want detailed specifications or precise tasks. This represents a huge divide in expectations. Often product owners or business analysts try breaking big ideas (sometimes referred to as “epics”) down into small user stories or job stories. Many expect product managers to design a prototype but that's simply another form of specification.

It's true: you cannot program to an epic. But having product managers write specifications isn't the solution either.

Tell a story about a problem; don't describe a feature.

For example, your customers may want your screens to adopt their corporate look-and-feel—using their color scheme and logo on screens and reports. For a product manager, “adopt corporate look-and-feel” is a reasonable requirement. But

developers often need to know more to begin working on tasks. Can we give them a color picker, or do you want us to somehow fetch the color palette from their existing web site? Does that mean using the client's logo instead of ours? Which file formats do you need to support?

Who in your organization can answer these questions? Is it the product leader? the product designer? the product team?

My recommendation is that product leaders own the *problem* while the product team (developers and designers) own the *solution*. That is, the product leader is responsible for the who, what, and why; the developers and designers are responsible for the how and when.

That's why the discovery meeting works.

Ideally, the team will discuss the problem and propose a few solutions. The team will weigh the relative merits of each solution and the product manager will guide the team to choose a solution that's appropriate for the persona and problem.

The discovery meeting idea is predicated on having a cross-functional product team with experience. If you're working with low-skilled programmers, someone—perhaps the product manager—will have to give them tasks.





## CHAPTER 4: CREATE

*“Perhaps the main reason we were behind schedule and over budget was because budgets and schedules are based on previous experience with similar projects. We really didn’t know how much it would cost to build or how long it would take.”—  
Tom Kelly, Grumman Aircraft (now Northrop Grumman), on building the lunar module for the Apollo program*

Why do we have schedules? There is often a false belief that schedules *force* teams to deliver; they force teams to be accountable. But schedules cannot force productivity; schedules cannot force creativity.

Most important, schedules do not allow you to complete more work than is possible.

Sometimes it seems each department thinks everyone else’s jobs are easy and that we would all be more successful if

“those other guys” would just work harder. Maybe that’s why everyone wants schedules and roadmaps.

Plans that fix scope, resources, and dates simply don’t work. The logic behind Kanban and other process methods gives us a better chance of success.

## Product Team Productivity

It seems that neither product leaders nor executives truly understand the mechanics of developing technical products such as software. You put together a team, the team works on requests, they work from a list in priority order, and they produce output on a consistent basis.

Or do they?

A product team is a group of individuals—developers, designers, testers. They’re people, not machines. It takes time and leadership before they can work together as a team. They must learn to communicate; they have to learn the product basics so they can safely perform updates. They need to build comradery.

A team’s productivity remains fairly constant until you change the composition of the team. **Every team change reduces its productivity.** New people, new methods, and new tools will all impact team productivity negatively in the short-term.

Every time there’s a change in the team makeup—whether adding or removing people—you change the team dynamic and its productivity. Unfortunately, many development leaders reassign team members to other projects frequently, moving

people from one team to another, as if developers were interchangeable parts of a machine. And each change reduces the team's productivity. Even *adding* a person to a team usually slows it down.

Once a team is formed it should be left alone—that is, if you want to maintain their productivity. Healthy product teams rarely change members; healthy teams minimize distractions so they can remain focused on their tasks.

A sales rep had an opportunity to pitch the product to a major client and called on the product team to build a special demo using client-specific examples. The team was glad to help out, but no one seemed to realize that the developers were now selling instead of developing. Long after the demo was forgotten, the sales VP went ballistic when some roadmap items were pulled from near-term delivery. He didn't seem to understand that developers were not able to work on the sales demo and on product deliverables simultaneously. Code doesn't just write itself when developers are elsewhere.

Frankly, in this case, neither the development leadership nor the product manager did their jobs. They failed to communicate the ramifications of the decision to redirect limited development resources to a sales-related issue.

In another company, the Human Resources department was planning their annual charity drive but unfortunately, they scheduled the kick-off event during the week before a major software launch. The product leader sent a note to the head of HR saying the product team would not attend. As you can imagine, HR threw a tantrum about their goal of 100% participation; however, the product team had a goal of shipping a new release and didn't have time for an all-day corporate

meeting. The product leader said, “Just renew whatever we did last year, and call that 100% participation.”

Corporate meetings and other interruptions impact productivity. A good product manager gives developers permission to ignore directives from elsewhere.

Honestly, most teams should plan only a few hours a day of productive time and use the other work hours for meetings, research, email, and other mundane but often necessary office tasks.

## Inspire the Product Team

Good managers inspire product teams; they don’t mandate deliverables.

The more context a product leader can provide—the personas who use the product and the problems they face—the more the team can use their creativity and judgment to solve problems in innovative ways.

The product leader kicks off a new series of work with a team briefing and discovery session. In the team briefing, you’ll share your business goals for the product and then get into a detailed discussion about personas and their problems, as we saw in the last chapter.

In my first product management role, I was led to believe that developers were “factory workers”—they just wanted to be assigned work tasks. Imagine my surprise when they asked me for details on the current and projected revenues, what marketing campaigns were planned, and how the product fit within the portfolio.

Many fail to realize that developers and engineers are craftsmen and -women. They are proud of the work they do—and the measure of their success is satisfied customers. They love customer anecdotes, particularly of those customers who are using the product successfully. Many sales teams ring a ship’s bell or buzzer whenever they win a deal. One product team copied this ceremony and rang their bell after completing a successful product release or market launch.

Give the team a prioritized list of work—often called a backlog. Work on the most important things, one at a time, in priority order. Whenever a defect is reported, it’s likely to go to the top of the queue. And yes, that means some functionality won’t get built.

For developers, products are like their children. They’re proud of the product. And they want to hear about their successes. So, start all product team briefings with a success report: anticipated revenues, competitive wins, and successful deployments.

## **Feed Your Team Using “Pull”**

There are two approaches to delivering stories to your product team: push and pull.

Typically, product managers and executives “push” more and more requests to the developers. Even though the team already has a bunch of things they’re working on, we push another few and then another few and then some more until the team is paralyzed with unfinished work.

In more practical terms, the product manager builds a long list of prioritized requirements, perhaps grouping them into themes or releases, and sends the team a few hundred items at a time.

This is probably the most common scenario despite its serious negative effects.

A long list of stories is overwhelming to the team, which damages both their spirit and their productivity. This approach also tends to deliver a bunch of disconnected stories, so you have 80% of everything but not 100% of anything. Most important, pushing a lot of work at one time means you can't reasonably re-prioritize the work when business conditions change.

Rather than pushing a bunch of work to the team, I prefer to let them “pull” work when they're available to take on new tasks.

In this scenario, the product leader maintains a comprehensive list of prioritized work—a backlog. Whenever they have bandwidth, the product team requests more work and the product leader hands them the next thing from the prioritized list.

Since the product leader is constantly prioritizing the list, the most important items are always on the top of the queue.

In one product management job, I was told the product team was terrible; they couldn't seem to finish anything. Investigating further, I found the problem wasn't with the team but with the executives. They were changing the priorities constantly. They didn't seem to realize that one sentence from the leadership team meant weeks (or months) of work for the

product team. I found the developers and designers were quite competent but, with an ever-changing priority list, they couldn't complete anything before a new top priority interfered.

We switched the team from a push approach to a pull technique. We never messed with anything that was in progress. All new ideas went into the product queue, were prioritized based on business value, and then pulled by the team as their bandwidth allowed. Using this technique, an executive's pet project could be put on the stack and then pulled by the team when that item became a top priority.

And when you're working in short cycles, new important items can be started every week or so. It's not like we have to wait months or years to begin a new initiative.

## **Prioritize for Business Value**

A product's feature list (or backlog or story queue) is constantly changing. You need to have an easy method for prioritizing (and purging) so your queue doesn't become a garbage can.

When Meghan was hired into a new company, she found hundreds of items on the developers' to-do list. Some items were over a year old. She and her team quickly went through the list and simply eliminated many items that were no longer relevant. They next identified "must have" features that could be developed and delivered within 90 days. Each quarter, they revisited the list to purge low priority items and select high priority items.

Here's a set of "yes or no" statements that can help you score each Product Story quickly:

- Most of our customers will use this
- Industry regulations mandate this
- We lose deals because we lack this
- We will close more deals if we have this
- We will be able to charge a premium for this
- This will improve customer retention
- This will reduce our costs
- This will improve our business model
- We can deliver this quickly
- A customer contract requires this
- This is an executive priority

Some of these statements are focused on your business needs; others on market needs. And this list embraces the reality that sometimes you must deal with mandates from executives and standards bodies.

You can add more items to this list and you can factor each answer differently. The total number of statements isn't particularly relevant as long as all stories are ranked using the same set.

The easiest technique is to score one point for each yes answer. Add 'em up for a story score.

But some of these questions could have a higher value depending on your company focus and the product's lifecycle stage. You might decide "We will be able to charge a premium for this" and "This will reduce our costs" ought to be scored higher than the others. Just sum up the responses for a score.

Use this rating scheme for each story and you'll have a long list of prioritized items. You'll add new items continually



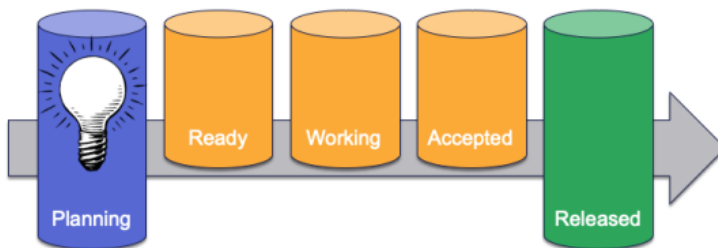
using the same technique and always have a prioritized list of items.

You'll realize the true importance of an objective rating scheme when you get to building the product.

## Product Stories and The Five Queues

To implement the “pull” approach, we leverage a series of staging queues where each product story moves from one state to another, from planning to released. This is based on the Kanban principles for managing and improving the flow of work.

Using these queues, a product leader can tell at a glance the status of any story. Furthermore, the product manager will ensure that only a few items are in the Development (or In-Progress) queue. For this technique, you can use a cork board with index cards or various automation tools.



© 2012-2018 Under10 Consulting

## Planning

All ideas begin in the Planning stage. Here, you'll attach personas, problems, critical dates, outcomes, needs, constraints, and other relevant context as you flesh out the story. This queue will have the longest list of stories.

## Ready

The Ready queue is for all stories that are ready for development but not yet pulled by the team to begin work.

Stories are moved from Planning to Ready after a formal discovery session. That is, your product team agrees that these stories are ready to be pulled into development.

In this queue, you can adjust priorities continually and provide more context up until the story has been turned over to the product team.

To feed the developers, you'll consider only the items in the Ready queue. These items are prioritized for business value but always take into account your own judgment. Just because something has a high priority, it may not be the most important thing to work on next. Look at your ready queue from the standpoint of "What is critical to be done in the next 90 days?" Or, "What must we do to survive?" These items are the ones you'll want at the top and ready to pull into the Development queue.

## Development (or In-Progress)

The Development stage is only for work in progress—all stories being actively worked by your team currently. Using the “pull” approach, the product team pulls work from Ready to Development only when they have the resources to take on more work.

Ensure your team is cautious to limit work in this queue—the team can easily get overwhelmed by too much work or paralyzed with indecision on what to do each day.

Once a story is in development, the product manager cannot manipulate it. After all, you don’t want your team trying to hit a moving target.

## Accepted

The story moves into the Accepted stage when all its required elements—needs, constraints, and outcomes—have been satisfied.

For those who release work on a scheduled basis, you’ll build up many items in the Accepted queue and formally launch those stories into the market. For an organization with continuous deployment, the stories could be made available instantly. But once it’s accepted, you cannot change it. After all, development is done.

Once a story is accepted, the story sits in the Accepted queue until it is publicly available.

## Released (or Delivered)

When the story is available to the market, you move it into the Released queue. Some teams prefer to call this queue “Available” or “Delivered” since the team has delivered the capability to customers.

You’ll see in the next section the power of these queues as you build and deliver products. These queues are also reflected in the public roadmap.

## Formalize Story Acceptance with a Demo

The team pulls work from the Ready queue; the team finishes the work and calls it “done.” But is it?

Work is not done until it has been formally accepted by the product leader or product owner.

If you’re using the queuing method just described, you know what stories are in development. And they cannot leave the Development queue until you’ve seen them working. That is, a story moves from Development to Accepted only when the product leader has determined that all of its required elements have been built and tested.

When the team agrees that the story is done, they demonstrate each completed story to the product leader and other stakeholders. Some call this a “demo.”

As we saw before, needs, outcomes, and constraints define the business rules for a story: it must have this; it cannot have that; it must do another thing. The demo will show that all aspects of a product story have been completed.

Some stories turn out to be pretty big so the team may challenge you to accept a portion of the work and move the remainder back to Ready. That's okay as long as the completed portion of work can be safely released to customers. Your goal in acceptance is to make sure you've solved a complete problem or scenario.

I strongly recommend a formal ceremony for acceptance. Ideally, the product team will fully demonstrate the story so the product leader can accept or approve the completed work.

Also take time to congratulate the team on their work—everybody likes a sincere praise after all.

Here's the good news: formalizing this workflow ensures that the team and the product leader are working to common goals. And also allows the product leader to share what's working with other stakeholders.

When all aspects have been completed, the story can move into the Accepted queue to await release to the customer base.

## **What About Defects?**

Rating defects is often a nightmare. Bad developers rate critical bugs as enhancement requests, so they don't have to work on them. Bad QA managers can use bug ratings to make a political point. Bad salespeople tell their customers to rate all enhancements as a high-severity defect to ensure the desired feature gets delivered.

One company instituted a new rating system: "We're 'customer driven' so let's have customers rate the severity!" Seemed like a good idea at first. But within days, there was a

new field in the trouble ticketing system: “real severity.” So many customers had rated their feature requests as critical that the rating system broke down; customer support soon set up a new field for realistic severity ratings.

You’ll want to implement an objective rating system for defects. A common technique is to rate defects by severity using a 4-point scale, like this:

- Severity 1 (Sev1): system outage
- Sev2: critical functionality not working
- Sev3: non-critical functionality not working
- Sev4: not working as desired or enhancement request

Generally, a Sev1—system outage—must be fixed immediately. In Kanban, this is called “Expedite” and bypasses all other work in progress. The product team literally stops whatever they were doing to attend to the expedited item.

Sev2’s and Sev3’s are quickly discussed and moved into the Ready queue where they’re prioritized against other work. Sev4’s will be put into the Planning queue for future consideration.

Consider an accounting product. A customer wants to have a running total column on the checkbook register. How would you rate this? It’s not a feature that is failing but a new request; therefore, it’s a Sev4. Still, it’s a good idea so we’ll probably put it high up in our backlog.

Another example: a report doesn’t total correctly. Is that a Sev2 or a Sev3? In this case, it depends on whether the report is critical or not. If it’s a critical daily report used by most of the customers, then it’s a Sev2; if it’s a non-critical report that isn’t run often, it’s a Sev3.

Think of it another way: Do you need to stop development of planned work to fix this? Does it affect your company operations? Does it affect the customer's operations? Does it affect many customers or just one? Is it worth delaying the next release?

This triage can typically be done by customer service and quality assurance without involving development or product management. Yet one team spent two and sometimes three hours daily debating the severity of reported defects. What a waste of time! Better to classify them as they are created and let the scale determine which will be addressed now or later.

In the end, severity rating shouldn't be a major daily distraction. Use an objective system so your teams are always focused on the most important work.

There will always be bugs. After all, developers are human; sometimes mistakes are made. Our products are increasingly complex and it's sometimes hard to see the impact a change might make elsewhere.

## **The Five Queues Determine the Product Roadmap**

The best public roadmap is an accurate reflection of your plans. The closer you are to finishing a set of work, the more confident you are in committing to delivery.

Remember the five queues? As you learned previously, stories move from Planning to Ready, Development (i.e., In-Progress), and Accepted, and then on to Released.

Now, to make a roadmap, just reverse the queues.

Recently Completed	Soon	Next	Future
Update roles, assignments, and permissions	Launch Plans 2.0	Playbook Today	Refresh persona definition, standard fields
Activate the purchasing of plans	Product Stories 2.0	Edit/CRUD 2.0	Integrate to JIRA
Formalize movement of Product Stories and Market Plans	Competitive Matrix Tool and Report	Capture, evaluate and assign ideas	Global search notes & attachments
System Status notifications		Website improvements	Provide a standard product pitch deck
Multi-product roadmap		Merge function: roles, assignments, and permissions	Archive old stories and plans
Prioritization Tool		Video quick tutorials	Easy input to product stories
Roadmaps		Battlecards and Competitive Intel	Customize for client's look and feel
Team briefings (internal)		Competitive feature comparison tool	Integrate with Salesforce
Artifacts and Guides		Easy edit for a web page	

With this roadmap view of the queues, items in the Planning queue are listed as “Future” since we have no idea when work on these will begin. Stories in “Ready” are shown in the “Next” queue—we’ll start working on those next, as resources become available. Stories from the “Accepted” and “In-Progress” queues are combined in “Soon” because they will indeed be available soon. And stories from the “Released” status queue are shown in the “Recently Completed” column.

It’s critical to remind everyone that no dates have been assigned to stories still in planning—we have no idea when we’ll begin work on these items so we cannot discuss when they’ll be available.

I’ve found it tremendously important to remind stakeholders—executives, salespeople, customers—how much you’ve accomplished. That’s why we emphasize the Recently Completed items in the roadmap.

One of my friends says, “Executives and salespeople have an enormous capacity for amnesia.” They seem so focused about what’s next that they forget about all you’ve delivered.



## Can You Develop Faster Than You Sell?

What if you could reduce the time it takes to get a story from idea to market. One technique is to do less more often. In other words, limit the amount of work in progress and release it frequently.

Here's a thought: What if you could have a product release (or two) during the sales cycle? How much would this help your sales team close deals?

For complex products, I challenge the product team to have a release schedule that is shorter than the typical sales cycle. Suppose it takes two months to sell a product to a new buyer. Just imagine what we could do with two releases in that timeframe.

“A buyer wants a feature that's shown in the future area of roadmap. What do I tell him?”

If it's crucial to a deal, we could choose to move it quickly through Planned and Ready queues and get it in the Development/ In-Progress queue. It's possible to have the feature online before the contract closes.

“Our competitor has announced a new feature.”

If you have a top-performing product team—and if it makes sense for your roadmap—you could develop and deliver the feature while your competitor is still showing prototypes. After all, your Ready queue can always be re-prioritized without impacting the In-Progress queue.

Even though some teams *can* deliver every two weeks doesn't mean that you must ship new features to existing clients. Release doesn't have to be the same as a launch.

You'll want to avoid the appearance of being a custom developer but the “pull” approach means your team can react to changing market requirements quickly.

## **Protect Your Product Team**

Another way to keep your team productive is to reduce distractions. Particularly when the product team is actively working on some critical capabilities, the product leader should help protect the team from outside interference.

My rule is: Never let salespeople or executives talk to developers. No good can come from it.

In truth, what I really mean is to prevent anyone from bypassing your process.

In one company, salespeople often popped in to see what the developers were working on. And frequently shared an idea or two about the deals they were working and the new requirements they were getting from prospects. The developers wanted to be good corporate citizens—and wanted to be liked—so they'd work on these sales requests (secretly) instead of doing the agreed-upon work.

In another company, the distractions weren't from the sales team; it was from the executives. On a weekly basis, generally just after the executive staff meeting, the VP of Development would come in and say, “Stop what you're working on; I have new requirements.” I arranged a sit-down with the president and shared my observations. “I found the problem: it's you. You're not getting new releases on schedule because the

requirements change every week. I can get us back on schedule if I can protect the team from interference.”

We did two things. We locked the salespeople and executives out of the development lab—their cardkeys simply didn’t work in this area of the building—and I gave the developers permission to say “No.” (Actually, what they said was, “Wow, that’s a great idea! See if you can get it on Steve’s roadmap.”)

If you want to ship a successful product, maintain a backlog of planned work. Add new work to the master list; don’t allow anyone to bypass the list. All new ideas are prioritized for business value and the list is continuously re-sorted. When the team asks for more work, you always have the top priority ready.

Let’s do less more often.



## CHAPTER 5: DELIVER

*“To do great and important tasks, two things are necessary: a plan and not quite enough time.”—Lucius Annaeus Seneca, Roman philosopher*

Meghan was hired a few weeks before an initial product launch. She couldn't find much core product information, so she was forced to start from scratch. She jumped right in to understand personas and problems so she could define the product positioning. And she planned a “soft launch.” She knew a staged approach is always more successful.

In Meghan's former company, the sales force was desperate for something new to sell and they put pressure on Meghan to rush a product to market. Furthermore, the executives were anxious to have some revenues to offset all the development costs. Everyone agreed a BIG SPLASH was the way to go.

But alas, the big splash launch was a failure.

It's a typical story for many product leaders. Despite her full slate of meetings, Meghan put together a presentation and demo script, and then trained the sales force—at least the ones who bothered to attend. Every salesperson made plans to show the new product to their top customers and needed someone to go along to support each sale.

In no time, Meghan's calendar was filled with more client visits than she could handle yet still the sales guys complained she was not supporting them.

On each of the client calls she could fit into her schedule, she realized the messaging still needed to be tweaked or some new sales enablement tool was needed or a few pretty important features were missing.

Sound familiar?

The “big launch” approach often fails because you haven't perfected the go-to-market materials. And really, how could you? You've been spending most of your time trying to get the product through development.

In my experience, companies rush to revenue before the product and its promotional tools are completed and tested.

For the DELIVER step, we'll provide guidelines for one or more product launches but first, a comment about roles and titles.

## **Product Marketing and The Promotion Team**

In the chapter describing the CREATE process, we saw product leaders or product managers working closely with a

product team (sometimes called a development team) guiding them with stories prioritized by business value. Let's use that same relationship for promotions. A product leader (sometimes called a product marketing manager) works with a promotions team, often called marketing or corporate communications, to introduce the product to one or more markets.

For product leaders, the question to be answered is, "What must I provide to the promotions team to help them get the product to customers?"

As with development, the product leader provides the who, what, and why while the promotions team provides the how and when. After all, your marketing people are experts in promotion—and product leaders usually aren't.

While product leaders will likely provide product and industry expertise to their teams, promotional execution of communications programs and campaigns is the responsibility of the promotions or marketing department. Product leaders typically have oversight on a product or market launch; they monitor overall launch status.

A launch plan introduces a new product or a new set of product capabilities, and ensures that other departments—the marketing, sales, and services teams—have the information they need to do their jobs.

## **Separate Release from Launch**

When the code is finished, packaged, and tested, the product is ready to launch, right?

Nope.

Remember this: a product *release* is not a market *launch*.

Just because the product team has finished creating doesn't mean the product is ready for delivery. You still need to prepare your company to sell and support the product; you still need to prepare your market to receive it.

Technical teams can release products but they never launch them.

Launches must be aligned with the rhythms of your market, not the rhythms of your internal teams.

Here in the US, few potential homeowners buy a property in November; they don't care that the house is vacant and the owner is anxious to sell. Instead, most home buyers start shopping in May to buy a house in June so they can move in July and put their kids in school in August. That means sellers need to put their houses on the market in April, not in November.

***Product release is the end of development; product launch is the beginning of marketing.***

That's why many organizations have one product leader (or a product owner) working with product teams to prepare the product for release while a different product leader (or product marketing manager) works with sales and marketing teams to prepare the product for launch. Product teams work forward to a release date; promotion teams work backward from a launch date.

To launch successfully, your sales teams must know which features are most exciting to your customers, how new features solve their problems, and how customers talk about the product and its features. Your sales and services teams will likely need



training on new product capabilities. You'll want to get a few early-adopter customers to put the product in production so the promotion team can write success stories and references.

Increasingly, the concept of a formal release is becoming obsolete.

And for many firms these days, both release and launch are meaningless. They simply release new functionality as soon as it is available. And they never launch.

Some online products, like Facebook, never launch, they just deliver new capabilities whenever they're ready. Others, like Squarespace, create launch events to show off new sets of functionality.

In the enterprise space, however, most customers don't really want continuous delivery. New releases are disruptive. After all, enterprises often put each new release through rigorous testing in their internal labs. They have to re-train their employees and partners. They have to communicate the changes to hundreds or thousands of individuals who rely on the product. They literally need to accept the release before putting it into production.

That said, software vendors with complex products typically have formal launch events, particularly when the products are sold through a direct sales channel. A launch inspires the organization's teams. A launch provides a tangible anchor for promotional campaigns. A launch creates a reason to re-engage with a lost client.

Your firm may do frequent or continuous deployment of releases but it's unlikely you'll do product launches more than a few times each year.

## A Compelling Release

You've planned a new release, pulling together ideas from the market and from the company. You think you've got a really strong plan but as soon as it's announced, the complaining begins. "When are we going to fix this capability?" "Where's my pet project?" "But I promised this feature to a prospect!" "Waa waa waa!"

Never forget, the list of features you're delivering is always influenced by political issues.

Some product leaders want each release to be purely focused on the needs of the market but prioritization explicitly says one thing is more important than another. Someone will always be disappointed because there are so many who want their ideas at the top of your list. To be successful, a good release requires something, however small, for everyone.

Here's one solution: Every product release should contain at least one thing for each of the product stakeholders.

- **Strategic for executives:** a feature that strongly aligns with your company vision or portfolio strategy
- **Sizzle for salespeople:** a new capability that strongly appeals to buyers
- **Productivity for customers:** a feature that improves usability or adds an often-requested capability. After all, people who pay maintenance or monthly fees expect some of their money to go to new stuff, not just for access and support.
- **Architectural for development:** a capability that makes the product easier to maintain or support, or one that provides a foundational element for future development. Sometimes this is about fixing clumsy code or improving security or reducing some annoying

defects. Or adding capabilities and diagnostics to make the product easier to implement, service, or support.

These are the items you emphasize in your roadmap and your internal communications.

One product team said they needed nine months to fix all sorts of internals and to address years of technical debt. But rather than just “going dark” for a year, the product leader and the product team blocked out the major areas that needed to be addressed and roadmapped them as architectural backlog items. They allocated time and money for new stuff as well as the internals. They met their internal problems and also their political needs.

Not every idea can be the number one priority. Release a set of functionality that has something for each of your major stakeholders and you’ll get better buy-in for your plan.

One thing for vision (and your executives), one for buyers (and your salespeople), one for customers (and your support organization), and one for maintenance (and your developers). That’s a good, solid release.

## **Prepare for Launch**

Executing marketing or promotional plans is usually not a product leader’s responsibility. Instead, the product leader maintains a dashboard which shows the status of each department for launch readiness.

At periodic launch meetings, each team member reports the status of his or her deliverables and you merely update your master to-do list or launch readiness dashboard.

What are some of the things that must be completed?

- Revise positioning and messaging
- Prepare sales tools
- Update all promotional pages
- Create announcement campaigns to prospects and customers
- Brief the press and analysts
- Train the sales and support teams

And so on...

The size of your company and its resources will determine the list of launch activities and when they occur.

A product launch also has goals, generally relating to revenue or adoption. The goals of a launch should be measurable—this many new users, this much revenue generated.

The good news is that you already know how to do this: you'll use the same techniques with the promotions team that you used with the product team. Do a briefing for the launch team with details about the product stories and personas, brainstorm the “features” of the launch (just as you did for detailed elements in your product stories), and rely on the experience of the launch team to elicit all the tasks to be done.

Write down everything you want to do, prioritize the list, work from the top of the list until you run out of time or budget. Just like product development.

## Refine the Positioning

The key artifacts that drive promotional activities are Personas and Positioning. You may have created these documents in the DISCOVER phase, but you'll need to create or refresh them as part of your launch planning, bringing in specific details from your planned product deliverables.

Positioning is a topic unto itself. In short, positioning is not the “elevator pitch” or a statement; positioning explains clearly who your product is for, how your product provides value, and why your product is different from alternatives.

How do you validate positioning? Listen to your early customers. As you continue to engage with customers, you'll learn what phrases resonate with them. You'll see which aspects of the product demo pique their interest. You'll perfect how you talk about the product. With these insights, go back to your product positioning and revise your writing to sync up with your new market understanding.

In addition to a tagline or “smile” message, you'll want to create 25-, 75-, and 150-word descriptions for use on the web site, in presentations, and in promotional materials. Write it down and get it proofed so you never have to write it again. Then use it whenever you need to send prepared text to anyone.

I call this “copy and paste marketing.”

Speakers and authors do this for their topics when they submit blurbs and short biographies to event organizers and publishers. They write it once and use it often. So should you.

For new releases, create two positioning documents: A refreshed document for the product overall and another focused

specifically on what's new in the upcoming product release. Your marketing team will use one for programs that encourage new customers to buy and the other when promoting to existing customers with reasons why they should upgrade.

I know a marketing expert who writes a 25-page messaging “book”—it describes the personas, target markets, and product capabilities plus the key phrases and descriptions used for each. This comprehensive briefing approach ensures the messages remain consistent no matter who is doing the writing.

## **Align Product Launch with Market Rhythms**

Aligning launches with market rhythms is a challenge for many product teams.

Agile methods are great for optimizing the development process and for delivering better functionality sooner. The key is not speed but a steady pace. With agile methods, we can see results as they're completed—instead of waiting until everything is in the final working state. That's what allows us to adapt to changing market conditions.

But most agile teams are agile in isolation. They dump new builds on marketing and salespeople—and customers—too frequently.

One team was working on a four-week development cycle and released new functionality monthly. Until the customers said, “Slow down! It's too fast.” I'm told most enterprise business customers want only two formal releases per year; others prefer, at most, one release per quarter. Look at Apple. Or Microsoft. How often do they put out a new operating

system or productivity suite? Generally, only one each year or every other year. Yet surely they're completing code every few weeks or so.

A sales leader explained that, for him, agile means that new releases *wander* into the customer base. The product owner notifies salespeople the day before and reminds them to tell their customers. "It goes live at midnight." They don't have formal releases any longer. They don't have launch meetings because they don't have launches. And the sales team can't catch up. It seems every few days a new set of features gets released into the customer base and the salespeople are faced with client questions that they're not prepared to answer. Where is the sales training? Where are the customer notifications? Where is the documentation?

Most agile product teams work to a 2- to 4-week rhythm. Most industries have a rhythm too. Education works around the traditional school year. Electronics generally are aligned with gift-giving seasons. You'll want to plan your release schedule with your customers' industry rhythms, not with your product teams' schedules.

Yes, it's possible to release new functionality every week or every month. The important consideration is not how often your team can *deliver* new functionality but how often your customers can *receive* new functionality.

Align your launch plans with your customers' schedule.

## A Phased Approach

If your organization wants to formalize product delivery, think in terms of phases: internal testing, availability by invitation, limited availability, and general availability.

A good example of a phased launch approach is Amazon's introduction of Echo in 2014. It was first available by invitation only to the top-tier of Amazon Prime customers—a group already inclined to be both supportive and enthusiastic. Once supply of the device increased and the messaging and promotions were perfected, all Prime members could order Echo and then in 2016, the product was available to any Amazon customer. This phased launch seems to have been hugely successful for Amazon. By the time it reached widespread availability, Alexa was already the top product in its category.

### Invitation Launch

You can use this same technique. Give early access to your best customers by invitation. Your best customers are inclined to be forgiving if and when they encounter problems. Some were likely participants in your beta program and early access is a nice incentive. They are inclined to give you extensive feedback. They are inclined to speak favorably about your efforts to others—including your potential customers.

However, an invitation launch is not a beta test.

For many vendors, “beta testing” has taken on another meaning: it's an early availability program for customers—and I think this is a mistake.



The key word in the phrase “beta testing” is not “beta,” it’s “testing.” If you look up phases of testing in a technical dictionary, you’ll find *Alpha Testing* is defined as testing of a release candidate *by vendor employees* on customer data and platforms. *Beta Testing* is defined as testing of a release candidate *by customers* on customer data and platforms.

Testing is necessary but it’s primarily a development concern; don’t confuse it with early availability. Both Alpha and Beta refer to testing the product’s *capabilities*; the invitation launch is about testing the product’s *promotion*.

By engaging with a select group of representative customers in an invitation launch, you will learn which features to emphasize and what messages resonate. You’ll learn more about the impact of the problems you solve so you can refine your pricing model. You’ll learn how they talk about the product so you can leverage that in your promotions.

Alexander Obenauer, founder of Mindsense, explains,

*As you move on to talking with the second, third, and fourth person, start to use their words as you describe it. Notice what sticks; what helps the concept click for the potential user more quickly. Keep those words. Keep iterating on the language using how ideal users and customers think about your solution, so that it clicks for the next person a little bit faster each time.*

Of course, it’s critical that these early customers represent your key personas. Otherwise their feedback has no meaning. A product for schoolteachers may also have value for facilities managers but an early acceptance program comprised only of facilities managers will not give relevant insights for your launches.

During the invitation launch, the product leader is both the primary salesperson and the support person. The product leader will “sell” the product to this select group and provide customer support as needed. Only in this way can the product leader see the product’s acceptance at every stage of the persona’s journey.

**Don’t let salespeople near the early access product.**

Salespeople always want to sell the newest thing (and demand that it be totally bug-free). And too often, they will try to sell to people who are *not* inclined to be patient or helpful.

One product manager had a meeting with a customer during the invitation phase—the feature set wasn’t finalized nor was the pricing model. Because of internal politics, she was forced to bring a sales guy along to the customer meeting. She explained their strategy, showed the prototypes and the planned roadmap, and led a discussion about the problems the product might solve in the customer’s environment. So far so good.

Then the customer asked, “How do you plan to charge for it?” The sales guy—who hadn’t said a word thus far—said, “Oh, I think we’ll charge around \$50,000 but I can guarantee you a 20% discount.”

“Oh,” the customer replied. “I really wanted to know if you were planning to charge a one-time license fee or charge by usage... but thanks for the discount.”

(Sigh)

An invitation launch is about validating the product delivery, not about selling.

## Soft Launch

Once the product and its delivery are ready for the market, initiate a soft launch, sometimes called a product preview. Roll the product out to one small geography or a subset of your customers. Make the new product available to a small sales group, perhaps only two or three salespeople. Then the product leader moves his or her focus from development to delivery, taking on the role of technical field engineer.

The product leader can't directly support a hundred salespeople, but he or she can support a few. Go on sales calls, observe what works and what doesn't, take notes on what the sales team and the clients are saying about the product and the problems it solves. Go for a meal after each call and chat with the salespeople about their perceptions and feedback.

Use the soft launch to perfect the go-to-market and sales enablement tools.

The audience most likely to adopt will be your existing satisfied customers, as we saw with Amazon's Echo. Expand availability of your products to a broader audience as your capacity increases.

And don't just think about capacity in technical terms, think also about capacity of your people. Sales, support, marketing, development, and product management teams all have capacity issues that are stressed when a product is launched. Make sure you're optimizing your resources with a soft launch.

## Global Launch

To quote my friend Saeed Khan, “You need to nail it before you scale it.”

Get your go-to-market right by working with a small sales team in a small geographic market, and then, when you’ve nailed it, you can roll the product out to the entire sales team with sales tools and messaging that have been perfected.

Be realistic about the resources necessary to address each geographic market.

It’s interesting to note that even the largest companies stage their global launches—expanding to only a few countries at a time. The sheer mechanics of coordinating availability of products in multiple markets can be overwhelming.

You’ve validated your marketing materials and sales enablement tools. You’ve trained all the sales and support people. Your web site, support lines, and sales channels are ready. Now you can launch globally.

## The Reverse Launch

While we’re talking about product launch, there’s a final format to discuss: The Reverse Launch.

When your product has stopped growing and the business retrospective reveals declining value, then you probably should retire the product. If so, it’s time for a “reverse launch.” Review your previous launches and just reverse them—take away everything you delivered. Remove web pages, cancel

incentives, remove price lists and discount schedules, and so on. Whatever you have done, un-do it.

The most important step in retiring a product is to stop paying commissions. Immediately. Salespeople simply won't sell products they aren't paid to sell. The last thing you want is a flurry of new customers right before you discontinue the product.

When you discontinue a product, recommend an exit path for your existing customers. Give them a free conversion to one of your other products or at least help them export their data for use in a different product. They won't be as upset if you offer good advice on finding an alternative.

Usually a simple email is enough. "We've made the decision to discontinue the product." And provide relevant dates such as when their accounts or data will be purged or support will no longer be available. Most customers are pretty understanding as long as their options are clear.

## **Launch Plans and The Four Queues**

We'll use queues to keep track of launch plan status, just as we did with product stories.

1. Planning		2. Ready		4. Completed	
Name	Priority	Name	Priority	Name	Released
Life of a product infographic	92	Roadshow	100	What's New	3-Mar-2017
Survey results infographic	90	Ebook for primary persona (part 2)	100	Customer success stories (3)	24-Feb-2017
Full survey report	82	Telemarketing campaign	95	Internal training presentation	6-Feb-2017
In-product promotional videos	75	Buyer assessment lead generation tool	85	Initial sales enablement tools	6-Feb-2017
Drip campaign for prospects	75	Ebook for primary persona (part 3)	82	Introductory sales presentation	13-Jan-2017
Paid search analysis	65	Customer success videos (3)	80	Press release	5-Dec-2016
Requirements for marketing automation upgrade	50	Industry survey	75	Website - initial content	24-Oct-2016
		4-part blog series	70	Web brochure	10-Oct-2016
				Message Guide	9-Sep-2016
				Marketing Brief	11-Jul-2016

3. Marketing	
Name	Target Delivery Date
Ebook for primary persona (part 1)	27-Mar-2017
Customer communications	31-Mar-2017

You'll likely have far fewer activities here than you did for Product Stories. The queues show launch plans in Planning, Ready, in Marketing (or In-progress), and Complete.

### Planning

All the plans that are not yet in flight are managed in the Planning queue. This means you haven't fully identified all the factors of a successful launch. Meet with the launch team to discuss and elaborate the details of the launch plan.

### Ready

Once you've led a discussion with your marketing team, move the plan to the Ready queue. You've specified the deliverables, promotions, constraints and goals, and set critical dates.

### Marketing (or In-progress)

Once you and your team are satisfied that all the issues have been documented, you can move the plan into the Marketing or In-progress queue. You'll check items off when goals are achieved, when events are completed, when campaigns are initiated, and when deliverables are, uh..., delivered.

### **Complete**

When everything you planned is done, you move the Plan into Complete status.

Some items you may want to mark as “Complete” when they are initiated rather than when they’re completed. For instance, if you bought AdWords for 180 days, you could check off the “AdWords” item once the transaction has gone through without waiting for the 180 days to pass.

## **Product Leaders and Launch Plans**

Launch plans are to the Marketing team as product stories are to the Development team. With Product Stories, we expressed elements such as outcomes, needs, and constraints but didn’t define tasks for individual product team members. With Launch Plans, you’ll document events, deliverables, promotions, constraints, and goals without defining specific tasks for individual promotion team members.

And as with Development, think of your promotions team as a group of smart, motivated people who are trying to do great things for the organization. You’re not their boss. You’re more like their customer or coach. And in the end, you want to leverage their advice and their expertise to deliver great products to your market.

The goal of a launch is to ensure you can connect with your market full of customers. You’ve prepared your sales team, you’ve automated your marketing drop campaign, you’ve nailed your positioning. Even in a reverse launch, you’ve

prepared the customer-facing part of the organization with what to say and how to say it.

One company introduced a new product with a full-page ad in a major technical publication. They did almost everything right in the launch plan. However, they dropped the ball on one thing: they used the company's toll-free number in the advertisement but didn't tell the operators about it. On the first day, the bewildered phone operators handled hundreds of calls for the new product but didn't know what to do with them.

Luckily, they took messages. The product leader quickly found out and redirected all the messages to the sales team but much of the initial momentum was lost.

Launch is the beginning of marketing. It defines the campaigns intended to simplify connecting with buyers.



## CHAPTER 6: CONNECT

*“The organization must learn to think of itself not as producing goods or services, but as buying customers, as doing the things that will make people want to do business with it.”—Theodore Levitt, “Marketing Myopia,” Harvard Business Review, 1960*

In manufacturing, there is a role known as the “expeditor” who bypasses internal processes to get an order delivered to a customer quickly. Many product and marketing managers are expeditors; they’re constantly bypassing processes to achieve a short-term goal. Salespeople tend to be this way too.

“Do whatever it takes to get the product to the client” is a nice sentiment but if *every* order must be expedited, your process is broken.

Good product leaders have a knack for finding patterns and addressing them with sales tools and customer-focused

collateral. And great marketing managers use phased launches to let people know they've solved a pervasive market problem.

According to a 2019 survey, 46% of a typical product manager's schedule is spent supporting individual salespeople. Because salespeople so often lack product and domain knowledge, and because the head of sales has understaffed the sales support team, we find many product managers are responding to proposal requests, delivering presentations and customer demos, and doing other emergency work generated by salespeople. I consider that time wasted—that's right: wasted.

Here's a common complaint: "But Steve, my bonus is tied to revenue, I *have* to help salespeople."

Yes, we want to help salespeople. However, the best way to support the sales team is not to help them *individually*; it's best to help them *systematically*. Sure, it's easier to just respond to their incessant requests—"I need a new presentation!" "I need a new brochure!" "I need a new sales tool!"—but the effective product manager looks for ways to help all salespeople using all the tools in the marketing arsenal.

After only a few customer visits, you'll see many ways to reduce friction in the buying and selling process: white papers to explain the technical underpinnings of the product; e-books to articulate the ramifications of industry shifts; standard presentations and automated demonstrations that show the top issues facing customers today. If you discover many salespeople can't quickly find the information they need, then an internal wiki or blog becomes an obvious solution. If your product is sold directly to consumers, without salespeople, you

already know that this same information needs to be readily available directly for buyers.

I like to think of salespeople as distributors. They distribute both products and product information. Arming your sales teams with the necessary sales tools and customer collateral is one of the primary goals of a launch plan.

## **Create a Sales Playbook**

When my daughter joined a sales team, she was given a list of company names and phone numbers to call—that was the customer prospect list. When she asked for details about the product line, she was told, “Uh, there’s lots of good stuff on the network drive.” No wonder this company had a dismal success rate for their new salespeople.

That weekend, I worked with her to create a sales playbook. We created positioning documents for the company and its products; we created one-sheets for each product; we created a brief sales presentation; we created a few scripts for her phone calls. In effect, we created a paper version of a customer database and a sales portal.

The following week, she started using the sales playbook and her colleagues all wanted to know where she got it and how they could get one. And she made copies for everyone. (She’s so kind!)

Put yourself in the position of a new salesperson in your organization. What do they need to succeed?

- Clear understanding of the ideal prospect (i.e., buyer personas)

- Descriptions of the product(s) and how clients benefit (product positioning)
- Sales tools for each stage of the sale
- Competitive write-ups
- References of satisfied customers

Another way of thinking about what salespeople need to succeed is to walk through the buyer's journey. What are the steps each buyer persona goes through from awareness to purchase? Then simply provide customer and sales tools for each step.

In one product leadership job, I periodically booked time for open Q&A with my sales team. They asked me about the buyers; they asked about the domain and the trends in the industry; they asked about problems people were having before acquiring our products and how our customers succeeded after purchasing.

Invariably I received new insights on how to better prepare my sales team for success.

A number of times in my consulting I have learned that sales teams were unaware of the broad set of existing materials. They were produced but not used. In those cases, I formed a team of folks from product, marketing, and sales to inventory the materials and align them with the organization's sales process. They just needed to organize them into a playbook.

Interview a few salespeople to learn what's working and what's not. If you do face-to-face selling, go on a few client calls with the sales team and watch for friction. When does the potential client seem confused? Where does the team struggle? You'll often find yourself saying, "Oh, we need a slide for that." Or "I bet a one-page assessment would help here."

And that's the point. If you don't create sales tools, you'll be the sales tool!

## **The Importance of Sales Engineering**

Because many sales teams don't have a sales playbook or they lack deep understanding of the products, they often rely heavily on the product leader or product manager to help win deals.

One VP described the product manager's job as "Wikipedia for the sales force." But helping one salesperson isn't product management; it's sales engineering.

Sales engineers are the primary technical resource for a direct sales team. They know how to implement and use your products. They are frequently former customers, so they also know what's happening in the industry.

Sometimes called "systems engineers," "pre-sales support," or "field consultants," they act as the product expert during the sale, representing the technical aspects of how the product solves specific customer problems. They deliver presentations and they own the demonstration script for the product. They know how to "talk technical" with technical buyers.

If you have a complex product and a direct sales force, you need sales engineering. Sales engineers are the technical glue of a complex sale.

My friends in enterprise sales say they need a sales engineer for at least half of their prospect calls, so the ideal ratio is for two salespeople to share one sales engineer. Unfortunately, we often see four or five salespeople begging for assistance from one sales engineer.

And when they can't get a sales engineer, salespeople reach out to others—such as product managers, software engineers, and executives. Supporting individual sales events isn't their job—that's why you need sales engineering.

## Win-Loss Analysis

Do you have the right sales playbook? Are you running the right promotions? Do you have qualified sales teams?

Fundamentally, why do you lose deals?

For that matter, **why do you win deals?** Does anyone really know?

There's one special form of research that most product teams are neglecting—and they're missing a great tool. It's win-loss analysis.

Consider this: most high school sports teams spend more time and money on win-loss analysis than most companies.

You know what's really scary to executives? *They don't know why they win.* They get annoyed by losses, sure, but it really drives them crazy when they don't know why their products win. After all, if you don't know why you're winning, how do you know what you can safely change? What should you do more of or what should you stop doing?

And who better to tell you what you do right and wrong in selling than someone who recently tried to buy, successfully or not?

Smart product teams use win-loss analysis as a tool to document and triage the buying process. What steps does the

buyer go through when making a purchase decision? What information was used—and what was missing? How could you help buyers make a decision—hopefully a decision for your product?

One product marketing manager did a dozen interviews in her first few weeks on the job. She talked to recent clients—some deals won, some deals lost—and wrote a report detailing the product and process problems that were revealed. Unfortunately, her boss left the firm and was quickly replaced with a new VP. (Uh-oh, now everybody has to interview again for their current jobs.) She went into the new VP's office and saw her win-loss analysis report on his desk. He said, "This is the only document I've seen from this department that I trust. Everything else is based on opinions."

Hmmm, couldn't you make better marketing decisions if you talked to some people in the market? (You know, it's so crazy, it just might work.)

Here's a common question:

But Steve, don't sales reps do win-loss analysis?

Nope.

At best, sales teams do win-loss *reporting*.

I'm not down on salespeople here. It's not that they couldn't; it's that they *shouldn't*. Multi-deal analysis isn't their game. That's what product teams are supposed to be doing.

Salespeople (correctly) focus on a single customer, not markets. A sales team's job is to sell more products to more clients. Salespeople should be selling, not managing the product and its promotion.

Product leaders should be looking for the patterns in *all* wins and *all* losses for *all* buying cycles, not just the results of individual deals.

One product manager found a pattern in all wins. Every enterprise deal the company won had one thing in common: the buyer participated in a non-customized demo during a visit to the corporate headquarters. Conventional wisdom was that buyers wanted a demo tailored for their specific needs, but the data revealed otherwise. Amazing!

The solution was to formalize the product demo as a key step in the selling process. The company hired a recent college graduate, gave him a room, and said, “It’s yours. Always have it ready for a demo.” He built a beautiful briefing center and delivered standard product demonstrations based on the primary messages and key features from the positioning documents. He set up a scheduled image restore that ran at midnight so each morning the demo setup was always pristine, no matter what was done to it the prior day. And he was always ready to do a perfect demo. After all, it was his actual job, not an interruption of his real work.

The company ran the demo center as a marketing program—with a dedicated staff and a dedicated facility. The company president credited the demo center with closing more deals sooner, substantially decreasing the typical buying cycle.

Do you really understand your customer’s buying process? A little win-loss analysis will tell you plenty about what your organization does right and wrong in promotion and selling.



Researching your buyer’s journey from interest to customer success is how to optimize your connections with potential and existing customers.

Learn what you’re doing right and wrong by interviewing sales teams and conducting win-loss analysis.

## Diagnose the Buyer’s Journey

A favorite diagnostic for the buyer’s journey is evaluating conversions, the movement from one buying state to the next. Work with your internal teams to identify measures for key buying steps, such as number of leads, opportunities, customers, and so on.

Consider this example from a marketing team:

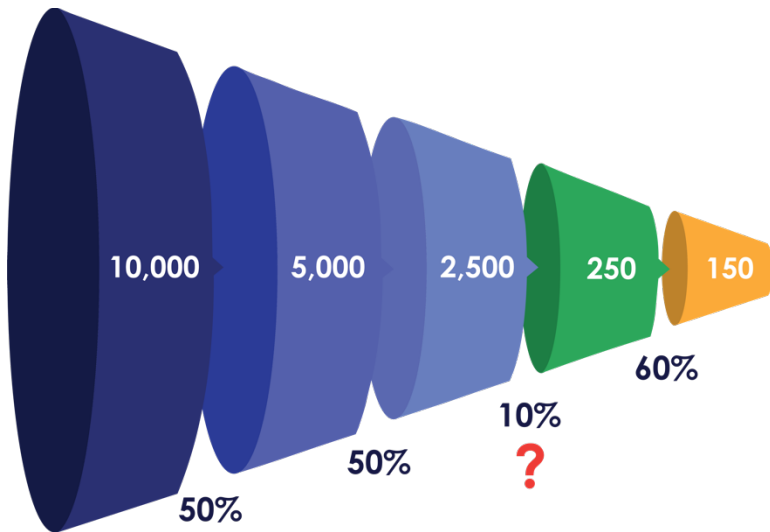
Stage	Activities	Number
Marketing Qualified Leads (MQLs)	Content, Events, Lead Nurture, Telemarketing	10,000
Sales Qualified Leads (SQLs)	Presentations and Demos	5,000
Opportunities	Evaluations	2,500
Customers	Executive Briefings	250
Referrals		150

Something is wrong here!

If your goal is 500 purchases yet you only got 250, it seems you just need to increase your lead generation activity. Double the leads, double the sales, right? Probably not.

What's wrong?

Let's look at the data another way.



When you look at the numbers as percentages, you see things fall apart during the evaluation. Only 10% of those who evaluate ultimately purchase the product. Before you start increasing activities in other areas, such as trying to create more leads, you need to see why your product fails during the evaluation. Maybe it's a product problem—your product doesn't have key features, or the user experience is clumsy. Maybe it's a messaging or positioning problem—you seem to get a lot of buyers excited with promises the product can't keep.

You can guess—maybe we need better onboarding or maybe our price is too high—or you can ask. Call a few clients and ask for them to walk you through their buying experience. Get the answer directly from the customer! That's why win-loss interviews and analyses are such powerful tools.

A simple and popular set of metrics is Dave McClure’s “Pirate metrics” (“pirate” because the acronym is “AARRR”). McClure measures these key steps in buyer conversions:

- Acquisition (Do customers find you?)
- Activation (Do they sign up?)
- Retention (Do they come back?)
- Revenue (Do they pay you?)
- Referral (Do they tell their friends?)

Failure to convert from Acquisition to Activation means that your promotions are working—after all, buyers are finding you—but your product descriptions don’t inspire potential customers to action. For example, “Click here to win a free car” is an acquisition promotion that will drive loads of unqualified eyeballs to your product information. They aren’t real buyers; they’re just trying to win a car.

Adapt these buying stage descriptions to your customers’ buying cycle and use the numbers to guide you. Any step with a large drop in conversions is an area the product manager needs to explore.

A lot of automation tools today are geared more to account-based marketing and selling which give you a micro view of the market. Look also at the macro view. Analyzing these conversion metrics will identify systemic problems.

## **Managing for Customer Success**

Do your customers love your product? Let’s find out.

Many go-to-market teams learn a harsh truth from their research and diagnostics. They often learn customers love the

sales team but don't love the product. These firms win a deal and move on to the next without ensuring that customer gets implemented.

There's a common method used to gauge the loyalty of customers called Net Promoter Score. It asks, "On a scale of 1 to 10, how likely is it that you would recommend our product or service to a friend or colleague?"

Authors Fred Reichheld and Rob Markey write in *The Ultimate Question 2.0*:

*Those who respond with a score of 9 to 10 are called Promoters and are considered likely to exhibit value-creating behaviors, such as buying more, remaining customers for longer, and making more positive referrals to other potential customers. Those who respond with a score of 0 to 6 are labeled Detractors, and they are believed to be less likely to exhibit the value-creating behaviors. Responses of 7 and 8 are labeled Passives, and their behavior falls between Promoters and Detractors.*

The Net Promoter Score is calculated by subtracting the percentage of customers who are Detractors from the percentage of customers who are Promoters, so the range of scores will go from -100% (where everyone is a Detractor) to +100% (where everyone is a Promoter).

As of 2020, here are NPS scores for some common electronics brands:

- +67: Samsung
- +47: Apple
- +45: Microsoft
- +15: AT&T
- +11: Google

These are all positive scores, meaning the percentage of Promoters exceeds the percentage of Detractors. Other vendors are not so lucky with scores near and below zero.

So here's the question: What is your ultimate goal? Revenue or referrals?

In our continual push for product growth, customer success often takes a backseat to revenues. We build a bunch of features to attract new customers but down-play the need for features that enhance the user experience.

Does customer success really matter?

“Yes,” says Jason M. Lemkin, self-proclaimed SaaS founder, enthusiast & investor. In his SaaStr Insider newsletter, he wrote, “I’ve now looked into over a dozen SaaS companies that stalled out somewhere in the \$20m-\$40m ARR range.” It’s simply this, he says, “The customers use the product because they have to, not because they love to.”

Why do growing companies stall? It’s not because of product features, marketing programs or sales leads. Lemkins says it’s because customers don’t love the product.

Organizations struggle when they focus on product revenue rather than customer success. To keep customers for the long-term, you need to achieve not just customer revenue but also customer success.

Ultimately, you want promoters who will recommend your product to their friends and colleagues.



## CHAPTER 7: LEARN

*“However beautiful the strategy, you should occasionally look at the results.”—Winston Churchill, Prime Minister of the United Kingdom from 1940 to 1945 and again from 1951 to 1955.*

You know the worst thing that can happen with a new product? You get a few customers. Not a lot. Not none. Just a few. And now you have a bad idea that must be supported and maintained and explained.

Meghan was hired to lead a new product initiative. Her first step was analysis of “what is”—so she interviewed developers, marketers, salespeople, support staff. She discussed the numbers with finance. And then she interviewed the market—both customers and non-customers. And quickly reached the conclusion: “This product is a Bad Idea.”

Based on this analysis, she determined the new product had cost the company just under a million dollars so far, and they

were only about halfway there. The salespeople didn't know how they'd sell it; the developers hated working on it; the services team worried they couldn't really support it. The product idea was materially different from the rest of the products in the portfolio.

Meghan called a meeting with the senior leaders of the company. She asked them, "Is this a strategic initiative?" The company either needed to set it up as its own division with a dedicated staff—or else they needed to kill it.

They killed it.

And the good news: although she had been hired to manage only this product, she got promoted to manage the entire product portfolio because she had made the best decision for the company—not necessarily the best decision for her career—based on market facts.

I challenge product managers to look beyond the product to the whole business—from idea to market. In addition, product managers should perform periodic retrospectives on the achievement of business goals. We're so busy *doing* that we don't leave any time to think. At the end of each quarter, we should take time to consider:

*What have we done right?*

*What can we do better?*

A product retrospective looks at the product capability, the best (and worst) customers, and the sales successes. What should we stop doing? What should we do more often? Let results drive your decisions for refining your product, your promotion, and your sales efforts.



## Market Retrospective

Is there such a thing as a bad customer?

Meghan hosted a customer advisory meeting of her top customers. One customer was particularly disruptive, focused entirely on his needs, with no interest in letting other customers discuss their challenges. After the meeting, Meghan complained about the customer to a sales manager who said, “Well, we gave them a screamin’ good deal and, uh, their sales rep probably made a few promises.”

Meghan went to Legal to see what promises had been made in the contract and was amazed. The client had paid less than the lowest discount (“because they’re a great brand name account”). However, the customer’s contract also stated they could never be used as a sales reference nor could they be mentioned in any promotional materials.

What good is a brand name customer if you can’t tell anyone?

Here’s a customer who is too demanding yet got a great deal—not a financial win for the company. Here’s a customer who would be a great reference—but refuses to be named.

Here’s a bad customer. We’re not making money and we can’t even tell anyone.

So Meghan did something radical. She canceled their contract.

Take a hard look at your existing customers. Consider their past payments and the costs you incur maintaining their account. Think about their customer support calls and their requests for new functionality. Which customers are simply too

much trouble? Consider dropping them on their next product renewal.

However, I remember a case from a company in Virginia. I proposed we discontinue a particularly difficult customer, but our head of support had a better idea: he hired the guy to work in our quality assurance department. After all, he had already proven that he could find every... single... bug... in our product. He went from being one of our most irritating customers to our most valued QA rep.

If customers are not profitable, why keep them? Cancel their contracts so you can focus your resources on good customers instead.

## **Business Retrospective**

A business retrospective is a periodic comparison of planned goals versus actual results. Before you define and deliver, it's important to determine how you will measure success. Setting these goals early in product planning is critical. Then when you get to retrospectives you know how you are measuring success, and you can adjust course if the results are not what you expected.

Perhaps the most important view of the business of the product is revenue by quarter for the last few years but don't limit your view to the financials.

Some non-financial metrics are:

- Are you attracting new customers consistently?
- Are you winning against competitive products?

- Is the adoption by customers increasing or decreasing? Price increases often mask a decreasing adoption rate.
- Are your customers using your product frequently? Look at how many users are active in a typical week or month.
- Are your customers achieving the outcomes they were looking to solve? Win-loss analysis should help you get this information.
- What are the trends in customer support? In other words, are trouble tickets and product complaints increasing or decreasing?

Even if you're seeing solid customer growth, compare it to the growth of the market to see how your market share is changing. Are you growing in sync with the market? Is the market growth slowing or is it your product growth that's slowing?

If your market or product growth is stagnating, it may be time to discontinue the product.

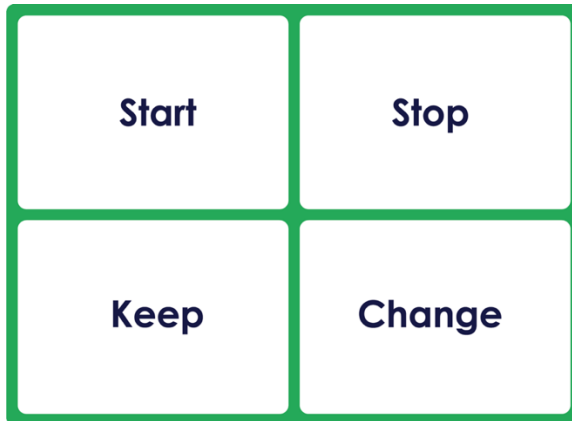
## Process Retrospectives

We've looked at the business and the market results. We should also take a look at our internal processes. Periodically examine what is working and what isn't so you can improve your internal processes.

You want to explore:

- What should we keep doing?
- What should we start doing?
- What should we stop doing?
- What should we change?

Bring your team together to explore processes and methods that you should keep, start, stop, or change. Draw a four-square grid on a flip chart or whiteboard and have your team members write in their ideas.



For each area, get your team to identify the top three things that should be done and then get a volunteer to lead the effort to address it. And try to drive them to a solution they can implement themselves without asking the other departments to change.

This simple tool can be used to review a launch, a release, or a method. Use it with your team to get insights on how to continually improve.

## **Always Be Learning**

Learning is at the core of everything. We learn from executives, colleagues, customers, non-customers, lost customers, prospects, industry leaders, analysts, and more.

Retrospectives provide a moment in time to reflect on what is working and what is not. Reflect on your personas. Reflect on your product functionality. Reflect on your promotions. Reflect on your processes. Then make decisions on what and how to improve!

Remember ABL: Always Be Learning.



## CHAPTER 8: ON PRODUCT LEADERSHIP

*“Because the purpose of business is to create a customer, the business enterprise has two—and only two—basic functions: marketing and innovation.”—Peter Drucker, American management consultant, educator, and author*

There’s a disconnect between academia and the rest of us concerning the word “marketing.”

In university, I learned that marketing was product strategy—you know, the four Ps: product, promotion, place, and price.

In the business world however, marketing is the name of a department. The common perception is that marketing is mostly about promotion and advertising.

What I have been describing is *product strategy*: defining products that solve problems for personas. That's what Drucker meant by "marketing and innovation."

In business, these questions are typically the responsibility of product leaders, often known as product managers.

The titles and activities in the category of "product management" vary from company to company. Put a dozen people in a room and you'll likely get that many different definitions of product management.

In startups, the CEO is the de facto product manager; in large organizations, there can be many, many people with variants of the title "product manager."

In recent years, I have watched those with the product manager title become more focused on the technology of their products and less focused on business aspects. They have become part of the development team rather than product-level business leaders. While most product managers are up to date on the latest development processes, they don't have a product leadership process to guide them from idea to market.

Product leadership requires business skills. Product leadership looks beyond the product development to its planning, promotion, sales enablement, and post-sales analysis. Product leaders manage the business of the product throughout the product life cycle and across the entire customer journey.

Over the years, I've had the opportunity to interview many company presidents, founders, and CEOs. Here's what they constantly think and worry about:

*How do we sell more of what we built?*



*Can we build what we've planned?*

*Have we planned the right products?*

Most company leaders see these as departmental problems, but I see these as *product strategy* issues—issues for product leaders. We can sell more by empowering our sales teams with more and better sales tools grounded in domain expertise. We can develop and deliver better by empowering our teams with product and technical expertise. We can define the right products with deep market expertise—the knowledge of personas and their needs, providing our executives with market-driven business justifications.

Perhaps the most common problem I see is this: understaffing in other departments leads to overwhelming demands on product management. We often find product managers providing extensive support to internal departments.

Developers are often kept in the dark about the market, so they rely on product managers for information about those who buy and use their products.

Because they frequently lack technical abilities, the marketing department wants product expertise from product management in the form of promotional content, product text, and screen shots.

Since salespeople often lack relevant job experience, they rely on product management for domain expertise. And since there are rarely enough sales engineers, product managers support individual salespeople with technical product information.

Alas, many product managers are actually *project* managers; they schedule resources, plan timelines, and report

project status. Some are serving as product *designers*. And some serve as *technical team* managers, spending the vast majority of their time with development activities such as estimating, DevOps, and assigning tasks to individual developers.

Even though these are important activities, they're not product management.

What we've seen is many organizations routinely overburden product management with responsibilities that properly belong elsewhere, simply because other departments have too few resources or inadequate skills.

## Defining Departmental Roles

*"In a well-run organization, each role has a single orientation; they either support customers or they support the market."—  
Peter Drucker, American management consultant, educator, and author*

Some company roles, such as sales and services teams, focus on customers one at a time. Other company roles, particularly product management, focus on markets—in other words, many customers at a time.

Visiting a customer with a sales or support objective isn't product management; that's selling and support. Customer (or market) discovery—that is, interviewing and observing several customers to understand their working conditions and purchase behavior—is product management, as we will aggregate the

information to understand the needs of an entire market, not just the needs of a single customer.

Defining the statement of work for an individual customer’s product implementation is a job for professional services—even if the work will include creating new functionality. Aggregating these requirements back into a future product for all customers is product management.

Sales, support, and services teams deliver products to customers, one at a time. Product management and marketing roles focus on problems for the market full of customers.

Here’s a simple way to look at departmental roles.

	CURRENT	FUTURE
MANY	PRODUCT MARKETING	PRODUCT MANAGEMENT
ONE	SALES AND SUPPORT	PROFESSIONAL SERVICES

We often find product teams broken into two groups: product *management* and product *marketing*. Product management focuses on future products and capabilities while product marketing focuses primarily on current offerings—the product and services we offer today. In both cases, their activities are targeted to a market full of customers: defining

and delivering the right products by empowering teams with understanding of markets, products, and goals.

Another way of thinking of these roles is this: product marketing is about ensuring growth for current products; product management is about planning future products.

## **Roles Through the Product Life Cycle**

*“The managers in your organization who have most consistently delivered results in the past may be the least skilled at delivering success in new-growth businesses.”—Clayton Christenson, The Innovator’s Solution*

As the product moves through its life cycle, from the startup phase to maturity, the activities of managing the product changes.

In early days, we struggle with honing the business strategy and achieving product/market fit. Here, working with the product team is mostly defining technical capabilities and validating basic functionality. When we move to the growth phase, our emphasis turns to scaling the promotion and selling, ensuring the field teams have the tools necessary for growing our customer base. Finally, in the maturity phase, we turn our attention to operational issues, optimizing for profit.

## During the Startup Phase

During the startup phase, the goal is to **optimize for learning**.

Every aspect of your product is a hypothesis. Who is the ideal target buyer? How do they buy? How important are the problems we're solving? Which are the critical features?

We can't assume any of our theories are true until they've been tested. What we're looking for is validation of the idea before we commit too many resources. What we learn may change the features we need, the price we can charge, or how we'll package and promote the product.

To optimize for learning, you'll need a testing mindset.

A technology firm developed a live video workshop on a specialized topic—one only a few people within a given company would need to understand. They used existing resources, leveraged their market reputation as well as the team's expertise and passion, and built the necessary content and systems. The new workshop was a nice addition to their portfolio and provided a competitive advantage.

They achieved their goal of learning the mechanics of a new delivery approach and were prepared to expand the product offerings in sync with customer demand.

Remember, every decision begins with an opinion but ends with market validation.

## During the Growth Phase

In the growth phase, we **optimize for adoption**.

During the growth phase, product leaders turn their attention to promotion and sales enablement. Presumably we have a workable product solution; we've achieved product/market fit. Now the fundamental question is: How can we sell faster? Or if you prefer, how do we improve customer adoption? We want to expand the share of market (i.e., more new clients) and expand the share of wallet (i.e., more sales to existing clients).

Consider the adoption of operating systems. Microsoft often struggles to get customers to move from one OS to the next. In 2016 Microsoft offered a free upgrade to Windows 10 in an attempt to get their customers to stop using older versions; by 2020, they had mostly succeeded. From a support standpoint, you can see the terrible costs of maintaining multiple desktop operating systems, yet the slow adoption clearly showed customers didn't perceive much value in the upgraded systems.

Compare the adoption rates of mobile operating systems from Apple and Google. Apple customers upgrade to new product releases quickly. Of course, Apple makes it easy since they control both the OS and the hardware. Meanwhile, users of phones based on Google's Android are often slow to upgrade, primarily because the phone companies themselves are slow to adopt. Android updates typically reach various devices with significant delays, often months after the release of the new version, or not at all.

Be cautious of large development projects during this phase. You'll want to avoid any special requirements that help one customer, favoring instead features that are valued by all customers. You'll want to focus on any feature that increases

sales but avoid massive changes that distract developers and salespeople.

As a product leader during the growth phase, stay focused on activities that improve customer adoption—and the ability to sell new units—in the short-term.

## During the Maturity Phase

During the maturity phase, you **optimize for profit**.

Look for ways to reduce both the cost of sale and the cost of goods sold. The goal here is to reallocate revenues from the mature product to fund new product initiatives.

However, the biggest challenges to managing a product in the maturity phase are the incessant demands to keep improving it. Even though the market is mature and new opportunities are shrinking, existing customers continue to request new functionality. Plus, the core product team always wants to tighten up the code and refresh the underlying technology. And the sales team needs “just one more feature” to close a deal. In short, customers, salespeople, and developers want to keep managing the product as if it is still in the growth phase.

The product leader’s role is to watch for signs that market growth is slowing, which signals the beginning of the end. Then, it’s time to aggressively optimize for profit.

Many companies have experienced the difficulty of moving clients from one product to another. One company went on an acquisition spree and bought four companies with products in the same category. The intention was to create one platform for

all customers except... the customers kept requesting capabilities on all four products and salespeople kept selling the discontinued products. Without strong product leadership, the company squandered its profits maintaining four almost-good products instead of creating the one excellent product their customers needed.

Some companies have perfected the maturity phase. They buy companies with products in shrinking markets and then do what the original company lacked the courage to do: they put the product on life-support. They stop all new development; they only fix major bugs. They treat the product as the “cash cow” it is, using the profits to fund new product acquisitions.

The maturity phase requires an analytic mindset to identify profit leaks, the courage to stick to the profit goals, and a complete lack of sentimentality. The product will soon be ready to retire; it’s time to let go.

## **Three Roles of Product**

Many people with the product manager title aren’t actually doing product management. Instead, they are prototyping and designing product specifications; they are prioritizing work based on the available skills in development; they are doing demos and closing deals. They are designers and development leads and project managers and sales engineers.

At consumer software companies, the product manager might be writing product specifications, running beta programs, and managing schedules. New product releases are



announced to the company like this: “The new build goes live tonight.” (Not much of a launch, eh?)

At one company, this was the norm. And chaos ensued as salespeople wanted to know how the company was going to inform the customers before the next workday. And of course, the marketing people didn’t know anything about the new release either; they got notified at the same time as the salespeople did. At 5:00pm. On a Friday. Before a holiday.

The product manager wasn’t doing product management, she was doing *project* management—for just for a release, not for a launch. She was too busy with the development team to spend any time with the rest of the company or the market.

At many B2B firms, product managers report they spend as much as half of their time supporting sales teams. With complex enterprise products, the sales teams often lack the technical skills and resources to support each sale so product managers spend time working individual deals and giving customized demos.

But while they’re doing what’s commonly called sales engineering, who is doing product management?

A little marketing, a little business, some technical support plus project scheduling—that’s today’s product manager. But what *should* a product manager should be doing?

The product manager title simply means too many different things to too many different departments.

I recommend these three new product titles aligned with the three phases of the QuartzOpen Framework: Product Strategy Manager, Product Planning Manager, Product Growth Manager.

## **Product Strategy Manager (“future”)**

Product Strategy Managers are first and foremost business leaders. They propose new products based on a deep understanding of the market, research and analyze the business and competitive aspects of new offerings, and empower other teams to create a suite of products and services that solve real customer problems.

They lead the DEFINE and COMMIT stages of the QuartzOpen Framework.

## **Product Planning Manager (“next”)**

Sometimes called product owners or technical product managers, Product Planning Managers focus on what the product team will deliver next. They are typically assigned to one or two product teams, serving as the customer representative in all team meetings. Responsible for maximizing the value of the release, they maintain and prioritize the release backlog.

They lead the DESCRIBE and CREATE stages of the QuartzOpen Framework.

## **Product Growth Manager (“now”)**

Product Growth Managers, often called Product Marketing Managers, are primarily focused on increasing revenue or adoption of the products already delivered—the ones the sales teams are selling today. They use their market expertise to

empower marketing and sales with content relevant to their markets, and they serve as the chief source of feedback from the market to the company. The Product Growth Manager leads sales enablement and go-to-market planning and should guide research efforts related to the buyer's journey and customer satisfaction.

They lead the DELIVER and CONNECT stages of the QuartzOpen Framework.

## **The Unicorn and the Purple Squirrel**

That's why it's often difficult to fill some positions; we are looking for a single person with the skills of three.

Many product management job postings are truly impossible to fill. Hiring managers seek to find someone with years of experience in product management... with a similar product... built on similar technology... in a similar domain... serving a similar market.

They're looking for a unicorn.

Who will work for peanuts.

Recruiters use another term for this: "Purple Squirrel." They're asked to find the perfect candidate, preferably local, with a technical undergrad plus an MBA, with experience in business and technology as well as domain and industry expertise.

As a hiring manager, think carefully about the skills you have on your team and the skills you need to support your products. Determine which areas of expertise are necessary to

support and grow your business. Then hire to the skills you are missing.

## **What Makes a Product Leader?**

*“An organization’s ability to learn, and translate that learning into action rapidly, is the ultimate competitive advantage.”—Jack Welch, chairman and CEO of General Electric between 1981 and 2001*

My father spent his entire business career at General Electric. In those days, the company manufactured batteries, light bulbs, toasters, turbines, jet engines, computing systems... you name it. But what GE really manufactured was... managers. GE perfected the business of business—creating standard processes and clear definitions of titles.

When my dad was promoted from engineer to supervisor, he went to the management training school in Crotonville, NY. There he joined dozens of his peers from all over the country; every person in the session had the same job grade and responsibilities, despite being from very different business units. Together they learned the job of supervisor. When he moved from supervisor to manager, off he went again to Crotonville to learn the mechanics of this next step in his career. And so on, through each step of management.

According to GE’s web site, nowadays the company invests about \$1 billion each year for management training at the John F. Welch Leadership Development Center.

Back in my father's day, job definitions were set in stone; perhaps they are still. But most of us don't work at GE. And our jobs aren't so clearly defined.

Product managers often seem to have an impossible goal. To be all things to all people. To be both strategic and tactical. To be involved in long-term strategy and short-term execution. To be expert in every aspect of the product. Product managers have deep expertise in domain, market, business, and technology—that's why they get pulled in so many directions.

According to one survey, product managers spend only 28% of their time on strategic activities.

Many product managers find themselves reacting to every issue and request, no matter how insignificant. They don't have adequate experience in the market to identify widespread problems to be solved; instead they rely on salespeople for market insights. However, for a sales guy, one customer opinion is a completed market study.

Too often, product managers are consumed by bug lists and never-ending feature requests yet lack a clear vision of what the product could be. They listen to what clients *say they want* rather than understanding what clients actually need.

Does this sound familiar?

Instead of defining product management as the “go to” resource for development, marketing, and sales, consider this alternative definition:

Product leaders systematically turn ideas into businesses.

They do this by:

- identifying problems shared by distinct groups of people,
- creating a vision of the future,
- inspiring and empowering their organization to deliver solutions, and
- monitoring key metrics and market feedback to refine their product vision.

It's not enough to have a good idea. Ideas are easy; execution is hard. Salespeople have good ideas; so do developers. The mechanics of turning a good idea into a business is where things get tricky.

Turning product and feature ideas into profitable businesses is the core of product management. Product leaders who drive product direction share these attributes:

- Product leaders are market focused.
- Product leaders are data informed.
- Product leaders inspire and empower others.
- Product leaders embrace a learning mindset and prefer facts over opinions.

*Product leaders* are not secretaries or gofers for others.

Product leaders lead the business of the product.

Start with personas and problems. Build something. Get it in the market. Get feedback. Analyze your metrics to determine what's working and revise your plans based on data from the right customers.

The objectives of managing products change throughout the life cycle so we need different skills in product management to meet these differing objectives.

## Your First Days as a Product Leader

*“The first principle of survival is not to worry about the impossible and concentrate on what’s possible.”—Robert A Heinlein, Have Space Suit—Will Travel*

You’re in your first days as a product leader. Now what?

There’s so much to do when you’re new to the job. Where should you begin? In no time, your calendar will be full, and you’ll have a zillion emails. Before the demands of others overwhelm you, prepare yourself to be the business and market liaison to the development and marketing teams.

When you’re jumping into a project midstream, start developing the core artifacts as soon as you can.

Suppose you’ve joined a team working with a long list of planned deliverables—an already over-full backlog. The team seems to be working okay and the last thing they need is more work, so start looking ahead to the launch. As you discuss the product with your marketing and sales teams, you realize you don’t have much information about the buyers and how they buy. Start studying customers and document your findings. Fill out a persona template with what you learn and keep updating it as you learn more.

For many new product leaders, their first days are spent in discovery: What are we building? Who is it for? What features should it have? What customers have contributed insights? What commitments have been made?

Ultimately, your role as a product leader is to make *business* decisions for the product, working from the best available information.

## **Refresh Your Domain Expertise**

If you've been in your industry for years, you probably have strong domain expertise but you may not be up-to-date on the latest information.

Perhaps the fastest way to get up to speed on your company and its role in the industry is to review the product and corporate presentations. Whether your company is a bellwether in the industry or on the periphery, what's been said in the past will help you understand how your company and its products are perceived, at least from the perspective of your new organization.

Take time to review the latest thinking in your industry. Catch up on the latest blogs and articles. Reports from industry analysts may reveal new industry trends, and perhaps show how your company and products are influencing them. Even if you've been in the domain for years, it's always helpful to take a new look from your new perspective as a product leader.

## **Fill Out Your Technical Expertise**

You may have some familiarity with the product from your past research. Now it's time to get into the raw details.



Get to know your new product. What information already exists for your product? You can probably find some customer documentation and help screens, release notes, product plans, sales and conference presentations, white papers and ebooks, and sales enablement tools. Review them all. Learn the key capabilities, particularly those that are competitive differentiators.

Review existing product plans and directions. Where is the product headed? Has anyone developed a roadmap for the next few releases? How does what you're seeing align with what you know about the domain and industry?

Understand the architectural challenges. Talk to the developers about technical issues for the product. What percentage of development effort is spent on infrastructure and defects instead of new functionality? And while you're at it, interview the developers about their perspective on your role in moving the product forward.

## **Update Your Market Expertise**

You likely have some market expertise, but it never hurts to give yourself a refresh.

Want to know what's going on with your product in the field? Ask customer support. They know about technical issues with the product as well as customer implementation problems. Sit in on some support calls and listen to customers directly.

Go on some sales calls. It's fascinating to examine the contrast between the product as perceived by the product team and by the people in the field. When you're on a customer visit

with your sales team, you'll hear how the product is being sold—whether right or wrong. You'll also hear unfiltered customers' problems in their own voices. Listen to the language they use; listen to the problems they're trying to solve. You'll get plenty of ideas for how to improve the selling and promotion of your products. And you'll get to know some salespeople in a more social setting; they'll be your contacts in the future when you need advice on sales enablement and product improvements.

Do some installation or implementation visits. For enterprise products, implementation is where all your sales and marketing promises meet the real world. Watch (or help) the implementation teams install the product; sit in on any customer training; examine closely the areas where the product must be configured or customized to work in the customer environment. You'll definitely get some great ideas on improving the product.

Eventually, you'll want to start visiting customers without a selling or support objective but get these initial customer touch-points under your belt first.

## **Leverage Your Process Expertise**

Now that you have a strong understanding of the domain, technology, and market, take a look at your internal product processes. What methods are used in your organization? Which of your favorite processes apply to your new situation? Where are the company templates stored?

Evaluate existing processes and systems. If you're stepping into an existing job, you'll likely find a set of methods that are already in place, formally or not. If you're part of a new product management team, you'll want to be a driver in defining your product processes.

What artifacts are necessary? What minimal set ensures success?

Start your own product playbook. Take your methods and the company's methods and put together a set of living documents. You'll want your product plan and financials, buyer and user profiles, positioning, requirements, maybe a price list or pricing model, and any other documents that you reference often.

The product management job is fascinating. It involves experience with business, market, product, process, technology, and more. You get to work with developers, sales and marketing people, finance, executives, customers, prospects... well, you get to work with almost everybody.

Re-orient your thinking. You're not here to *support* them; you're here to *inspire* them. To empower them with an understanding of personas and problems. Use documents and communication to help your teams help themselves.

## Standardize Your Team with Product Ops

*I get up every morning determined to both change the world and have one hell of a good time. Sometimes this makes planning my day difficult. —E B White, American author*

How do product leaders organize product teams for the best results? If your team is large enough, you should explore the idea of Product Operations (or “ProductOps.”)

ProductOps is a specialized role within product or product management that normalizes the function across all products and services. The ProductOps goal is to standardize and optimize.

ProductOps examines and standardizes processes with common templates and tools. It helps perform data analytics and acquisition and develops methods for meaningful engagement with customers and potential customers.

Here are some ways that ProductOps can benefit your product teams.

**Define roles and responsibilities.** Let’s have a single definition for each title and what they do. For instance, who should do win/loss interviews, analysis, and reporting? ProductOps can either coach teams in the best practices or perform certain capabilities (such as win loss analysis) as a service.

**Standardize methods and artifacts.** What templates do we use? How should we prioritize business opportunities? What’s the best approach for backlog grooming? ProductOps builds a “product playbook” of standard

templates and tools, adapted to the special needs of your business.

**Wrangle corporate and product data.** With so much operational data available, how can a new product manager make sense of it all? ProductOps can be the expert on data that's available and how it can provide insights to product decisions.

**Guide and curate market and customer research.** How do we set up customer interviews? Where do we store and share our insights? How can we execute experiments such as A/B tests?

**Evaluate and manage departmental tools.** What roadmapping tool is best? Do we store product information in Jira? How do we use Teams or Slack? Instead of multiple tools, ProductOps identifies departmental needs, evaluates available tools, and makes a selection. ProductOps then manages the tool and trains team members.

But there's a problem. The danger is that ProductOps may be perceived as responsible for doing everything that other product management roles don't want to do. ProductOps is a service to all product management roles but neither a master nor a servant.

The confusion of titles and roles is a problem for most organizations. Each department has pre-conceived notions of what a product manager should do. Instead of thinking, "do whatever it takes," let's identify the activities and artifacts that are the responsibility of your team members using QuartzOpen.

Create a team designed to serve your product and company needs using the new product titles described earlier. And make sure your team has the skills necessary to succeed in the job.

Create your own versions of the templates described in this book and use them to standardize your teams' processes. Ensure that all new ideas—from every source—go through a business prioritization process.

By examining at what works (and what doesn't) across all product teams, ProductOps identifies successful approaches and helps each team adopt the organization's best practices.

Product teams need standard tools, a common language, and training on product management best practices. ProductOps adapts industry practices to create an organizational “playbook” of methods.

In my first product management job, I was frankly surprised at the interest my team showed in the business aspects of the product. After all, I'd heard the developers didn't care about the business; they just want to code. My team surprised me. They asked me about the revenue goals and how the product fit into the company's strategic plans. They wanted to know the results from our marketing promotions. They wanted to know more about the people who bought our product and the people who used it.

I began conducting monthly meetings with the team to update them on what I'd learned over the course of the previous month: insights from sales calls and my direct customer feedback plus the effectiveness of our various marketing programs.

Soon after, I got requests from other departments—support, finance, professional services—to deliver my “vision presentation” to their teams.

I was no longer surprised. I had learned that everyone who works for a company is interested in its future! Aren't you?

Before you begin any project—a new product, release, or launch—give your team members a shared vision. We call this a briefing. It provides the context of the product—the buyers, the users, and problems that we're solving for them.

Let's all take a look at the forest before we focus on the trees... and the leaves.

In these briefings, you'll share the product vision and positioning, the personas, the priorities (and how they were determined) as well as the dashboard you plan to use for company updates. Also point out any executive "pet projects" so your team knows which are getting extra scrutiny.

Facts are the source of credibility. Interview and observe customers so you can provide both background and details for each problem. Develop your personas so we can stop developing and marketing to the generic "user" or "the customer."

After all, products solve problems for personas.

Become the problem expert on your product.

Use the Kanban queuing method described earlier to keep track of all stories and plans—those in planning, in development or marketing, or waiting to be released.

Conduct periodic retrospectives of the market, the business, and your internal processes so you're always learning and always improving.

Finally, I fear that many product leaders and marketing professionals aren't very good about promoting themselves. Be

vocal about your team's successes. Be public about the status of your products and promotions. Look for opportunities to share your roadmap or conduct a demo for colleagues. Write posts on internal systems about sales successes, marketing successes, and development successes.

And don't forget to share your customers' success stories with your organization.

I have embraced many best practices from leading mentors over the years and I continue to learn new methods. So should you.

Use what you've learned to turn your ideas into products—quickly and profitably.



## ABOUT THE AUTHOR

*“I have come to believe that most organizations need a small number of living documents, usually fewer than 10.”—Steve Johnson, author, product strategist*

Steve Johnson is an author, speaker, and transformation coach, focusing on modern methods for product strategy, planning, and growth. His approach is based on the belief that minimal process and simple templates result in a nimble product team.

Steve's work experience includes technical, sales, and marketing management positions at companies specializing in technology-based products. His market and technical savvy



allowed him to rise rapidly through the ranks from product manager to the executive suite.

Steve has been a long-time advocate for product management, with 15 years as founding instructor and executive at Pragmatic Institute as well as serving as an executive-level advisor to a number of technical product organizations.

In addition, Steve is the source architect of the **QuartzOpen Framework™** and is a member of the **Product Growth Leaders** network, a partnership of independent consultants focused on transforming product teams.

Steve was one of the first bloggers on product management. Since 1999, he has published thousands of articles and posts. His ebooks on product management and leadership have been downloaded over 100,000 times.

Steve is a popular keynote speaker at forums throughout North America, delivering highly relevant sessions for technology-oriented audiences.

Learn more at [\*\*www.under10consulting.com\*\*](http://www.under10consulting.com)