1. (Re)Defining Product Management

In this lesson, we start with the important questions: what is product management? Why build a product? What work goes into overseeing product development? What products does Christian use to keep his hair so nice? The essentials of product management are products! The device you're reading this on is a product, One Month is a product, Christian's shirt is a product. Products are anything that people use - physical or online, hardware or software. Products make business possible, which supports designers of products, and themselves make the production of products possible, which makes *more*... you get where we're going here.

What is a Product?

Something people can use.

Why do we build Products?

To satisfy other people's needs.

How do you get people (to work)?

By paying money.

How do we get money?

By selling product.

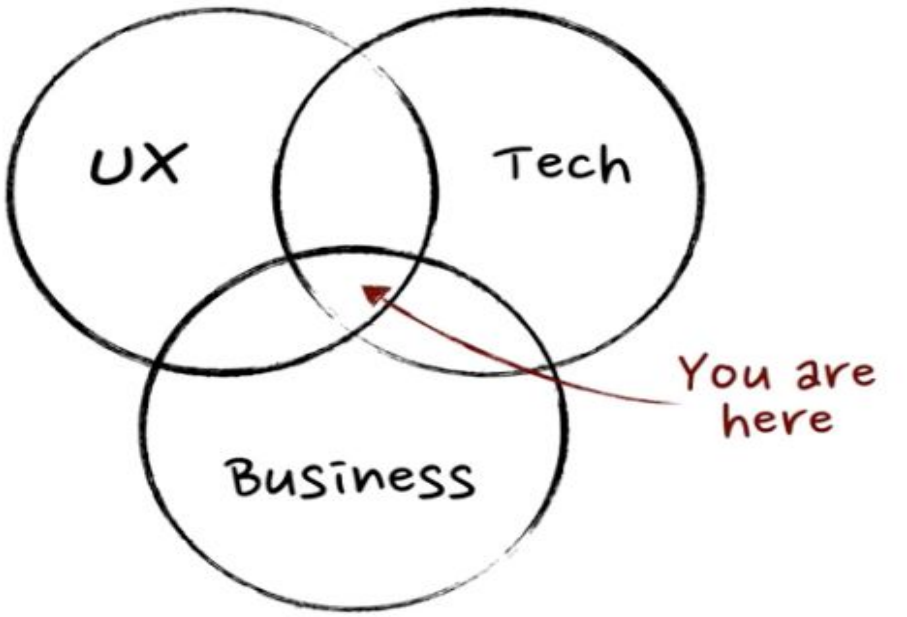
The above four lines are in themselves a cycle.

Products are sustained by the business (very important).

There are 3 components to a product -

1. **Technology :** What makes the product work. *Includes* - Software and Hardware engineers at work.
2. **Design :** What the product looks and feels like. *Includes* - Product and Industrial designers.
3. **Business :** What sustains the product. *Includes* - Business Development, Sales and Marketing.

Both managing or overseeing the launch of new products, therefore, means managing the intersection of Design, Business, and Technology. To put more visually:



Product management is a business strategy role.

Product management, then, is a role that requires big thinking about a lot of different aspects of a product and its underlying business. It is a business strategist position, keeping all the product and related business teams moving in the right direction.

In this course, we're going to learn how to manage the intersection of all of these components in order to make a product successful in all of them. Get excited.

2. Your Mission, Should You Choose To Accept It

Or, rather, should Christian choose to accept it a while ago at Shapeways. But in this lesson, we look at a specific example of a product management challenge in order to understand what a product manager does to address it.

Here's an example of a product management issue

The Challenge: Our customers want faster and cheaper shipping.

This is, remember, from the business side of things. Addresses this need would drive sales which would support the people who work on the product and enable them to refine or make more products.

Step 1: Talk to Engineering team about implementing faster/check-out shipping options.

The engineers say they can definitely add more options into the check-out process. It'll take X amount of time, but shouldn't be a problem to program.

Step 2: Talk to Design team about implementing the different options so that user can see the shipping options, and how they change the price/estimate when they're selected. They're also onboard.

But nobody at this point actually knows what they're making. That's where the product manager comes in. The product manager decides what actually is going to be built. What kind of shipping options? Which services? How many?

A product manager thinks of the questions that need to be answered to build out a new product, and then answers them. In doing so, you help define what the product is.

How do you do this?

Working With People

In the example so far, Christian's talked to business, engineering, and design teams. Now he loops in marketing to figure out what kinds of shipping options will actually useful for the users and be profitable for the company. He talks to the operations staff at the shipping factory to see what's feasible and what might streamline the process.

Product managers are editing all the ideas that they receive. The position involves careful listening and strategic thinking on how to transform all the different ideas you hear into the best possible and unified vision of how to go about creating the best possible and coherent product.

You need to be a **pragmatic idealist**: taking into consideration time, money, resources, and all the other real-world constraints and make the best possible product inside of them.

Lots Of Questions To Answer

- If we give customers faster and cheaper shipping will that increase sales?
- How much are customers willing to pay for faster shipping?
- How much cheaper do our customers want shipping to be?
- What are our competitors offering?
- If we give customers a cheaper shipping option how much slower can shipping be?
- Can we offer customers shipping at the price or speed that they want without losing money?

A product manager works with people.

A product manager works with people.

A product manager is a pragmatic idealist.

3. What is a Product Manager?

How do product managers determine the best possible course of action for a product's best possible outcome? Short answer: a lot of listening. Long answer: by owning the roles and not falling into any of the mistakes mentioned in this lesson.

So, thinking about the challenge Christian had to deal with - creating faster and cheaper shipping options - how would you go about sorting through all the ideas you'd hear from all the different teams, coming up with your own questions, and finding the best possible solution?

It may help to think about what a Product Manager is, and what a Product Manager is not:

A Product Manager Is:

- The mini CEO
- The strategic thinker
- The expert on the market
- The voice of the customers

The product manager needs to actually deal with and shape the product as though it's yours. Your job isn't to defer to the CEO, but to be the mini-CEO. You think strategically about the challenges of implementing a product, and use the expertise you have on the product's market to help shape and define how that implementation looks. But, you also have to make those determinations based on the perspective of the customers. It's your job as well to be their voice, to know what a really great product is going to look like to the people who're going to want to use it.

A Product Manager Is Not

- The boss
- The developer or designer
- A gopher
- Customer support

But you're not in charge. Your job isn't to tell everyone what to do, but to provide a unity of vision that's going to guide what everyone does. The difference is subtle, but it's there. You also aren't there to get lost in the weeds of specific design or developing specialization. That's what the designers and engineers are for, and so the product manager has to rely on those people who have the expertise to build out the product. Likewise, you're not around to run errands or do anyone else's job, including customer support.

More or less, you are the Great Curator: you listen to *everything* everyone has to say, and then determines what work needs to be done across the board to build the best product possible.

4.The Circle Of Life...Of A Product (The beginning, middle and end of a Product)

In this lesson, we cover the lifecycle of products - how it develops, grows, and changes - so that you as a product manager know where you are and what you're working towards.

The Six Stages Of A Product's Lifecycle:

1. An idea - coming up with what a product might be, what your customers want or need.

2. Development - building the product.

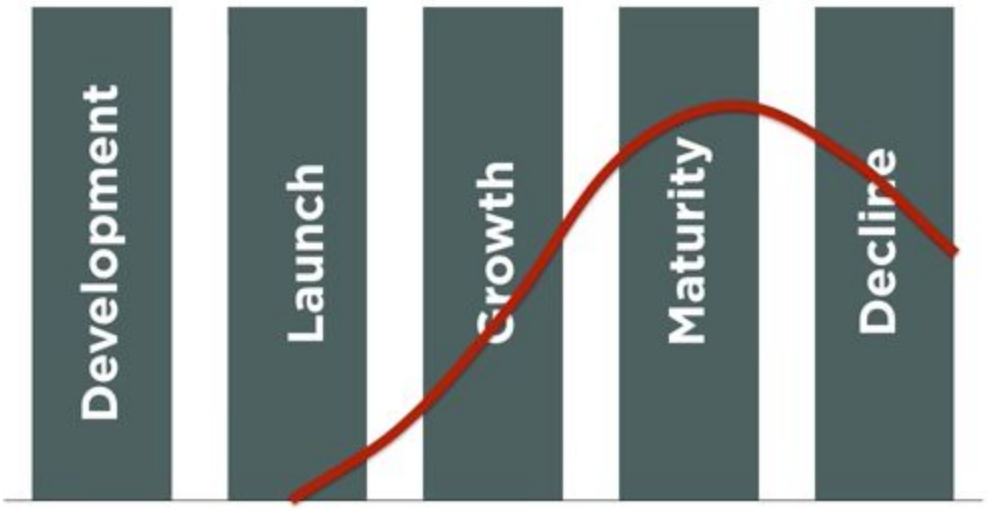
3. Launch - actually putting the product out there so that people can encounter it.

4. Growth - once it's out in the world, you need to go from no one using your product to (hopefully) a bunch of people buying and using your product.

5. Maturity - When the product's growth plateaus and it's less new. Sales can tapper off, or use of it can stabilize, without new users.

6. Decline - When your customer base decrease/decline, with people not buying it.

It can also look like this:



The nice thing about this is, that unlike our own lifecycles, people have control over a product's lifecycle. You can choose to do something about decline - and as a product manager, it's part of your job to help change or manipulate the product lifecycle so that either it, or the company, is growing as much as possible.

5.Reading The Product Life Cycle Signals

Knowing where in its lifecycle your product falls will help you anticipate its needs. So in this lesson, we go over the tools you can use and the signals you can identify to determine where your product is at and what you need to be focusing on. It's important to understand what stage of the product lifecycle your product falls into - conception, development, launch, growth, maturity, or decline.

Whatever stage you're dealing with changes your focus as a product manager. For instance, if the product is in its launch stage, your focus is on helping it to grow; if you're

dealing with a mature product, you're looking for ways to spur more growth. Knowing what step you're at is going to help you look ahead to where you need to be. And knowing where you're going is, as it is in most cases, going to help you be a better product manager.

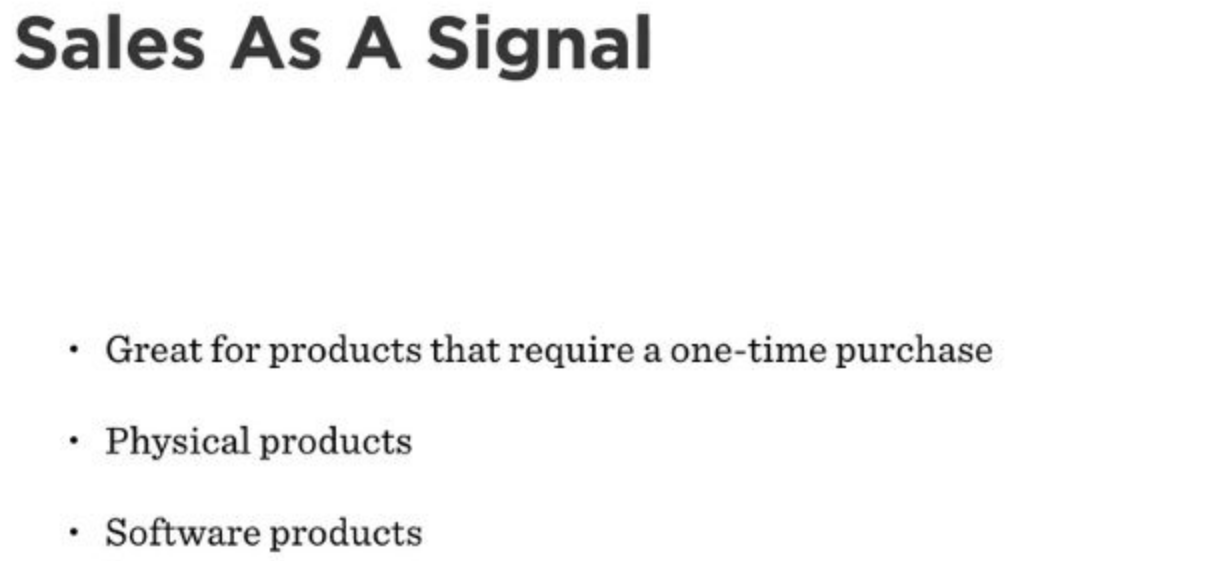
Some common signals that can help you determine a product's lifecycle stage:

1. Number of Sales - an especially fantastic indicator for one-time purchases. Are people actually buying it, whether it's a physical product or software.

2. Revenue - is the total number of sales multiplied by the product's price. But it can also be deceptive, because higher priced product sales don't necessarily imply growth.

3. Customers/Users - Customers are the people who buy your product, and users are the people who use your product. They can be one and the same (iPhone users bought their product and use them), but Facebook users didn't have to pay for the product they use. So looking at the number of users over time is a good way to track growth.

4. Rate of Growth - The percent of change from one period to the next. You can google the formula, but you have to pick the metric (the periods you're defining) you're tracking. It can also be misleading. If your user growth decreases, that doesn't mean you're declining. As long as you have positive growth, you're growing. When you have a negative percentage, that does indicate decline.



Revenue As A Signal

- Total number of sales multiplied by the product's price
- Revenue can be a devious signal
- Selling one $10,000 product and zero $200 products doesn't imply growth

Customers Or Users As A Signal

- Customers and sales are similar
- Unlike sales, customers or users might not need to pay
- Facebook, Twitter, and Instagram are great examples of that

Rate Of Growth As A Signal

- Percent of change from one period to another
- You have to choose a metric to track it's rate
- Also a devious signal
- Growth rate decreasing from 40% to 15% isn't decline
- Negative growth is decline

6. Seven Tips For Seven Phases Of Product Development

So you've been brought on to manage a product in its initial development phase. Great! How do you manage it? You work through these 7 steps, of course, however many times it takes to create a great product.

1. **Conception** - You may already have the idea for a product, but here you work on understanding the needs and wants your product addresses, and refining what your product is. This is the place to have your editor hat on, listening for problems, asking for feedback, coming up with and answering questions.

2. **Planning** - Here, you use the newly focused idea of what needs your product addresses in order to determine the exact shape/functionality of the product you're going to build. You should be discussing solutions to all those questions you came up with, looking for ways to improve on the existing competition, setting objectives for your product and coming up with a development schedule to execute your ideas.

3. **Building** - In the third step, you execute your plan and actually build a product. But this isn't a time for you to be hands-off. You should be documenting how the product will work so that everyone working on the product knows what they're doing, checking in with the stakeholders (the people who ordered the product built) regularly, and making sure you're adapting to the inevitability of things changing.

4. **Iterate** - In this step, you don't just get to stop and admire your shiny new product. You need to figure out what's working and what's not. You need to test what you've built, get feedback from your teams and your stakeholders, and ultimately fix what doesn't work.

5. **Launch** - Congratulations, you've made it to a whole new phase of the product's life! But this is only step five of development. You need to come up with a plan for the product's grand debut, figure out and probably be involved in the external communications that are going to let the world know about this awesome new thing you've created. You also need to make sure you're keeping all the teams involved with the product in the loop, so that they're comfortable with the understanding the public is going to have of what the product does and so that you can sell its best features. Don't leave anything to chance.

6. **Analyze** - Here, after you've cleaned up all the champagne from the launch party, you can step back and answer the biggest question: is the product initially successful? How is it doing? Make sure you share the results of your analysis with your teams so that they have context for what they accomplished, and know how to maybe improve on

the next product. Also, this is the place you can identify opportunities for the product going forward, identifying what might come next.

7. **Deciding**- Now that your product is standing on its own two feet, you have to decide what to do with it. You may need to grapple with how to maintain it if it's meeting the goals you set for it; or maybe you need to make the call and choose to kill it, if it isn't working. Based on your analysis, you need to determine how to re-invest in it to enhance it and create more growth for it.

Remember that your product may very well change - and if it does, you shouldn't stay tied to the gameplan for the old version of the product. Don't be afraid to start over and begin at the beginning of the steps. It's the very best place to start. Work in whatever order makes sense for your product. But **don't** skip steps. All of these things are essential to helping you create the most successful product you can.

Conceive

Discuss the idea for your product, and refine it.

- Listen for problems, ideas, and feedback
- Answer questions you might have to help defie the what
- Refine the idea of the product
- Determine what the product will do and what it won't do

Plan

Use the focus of your idea to determine the product you're building.

- Discuss different solutions that you can build
- Learn from competitors with similar solutions
- Set and share objectives, hypothesis, and goals

Build

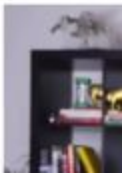
Execute on your plan and build a product.

- Start building the product with your team
- Document how the product will work
- Regular check-ins with your team and stakeholders
- Adapt to change

Iterate

Figure out what's working and what's not - and change it if necessary.

- Test what you've built so far
- Get feedback from your team, stakeholders, and customers
- Find out what doesn't work
- Fix what doesn't work



Launch

Make your product accessible to the public.

- Create a launch plan with your team
- Help plan external communications
- Don't forget about internal communications
- Don't leave anything to chance

Analyze

Determine if the product is successful.

- Analyze how the product's doing
- Share the results of your analysis
- Identify opportunities based on your analysis

Decide

Determine the future of the product.

- What's next for the product?
- If the product is meeting it's goals, maintain it
- Re-invest in the product to enhance it
- If the product doesn't work, fix it or kill it

7. Would You Like To Play A Game? (Turn Customer Needs to Customer Ideas)

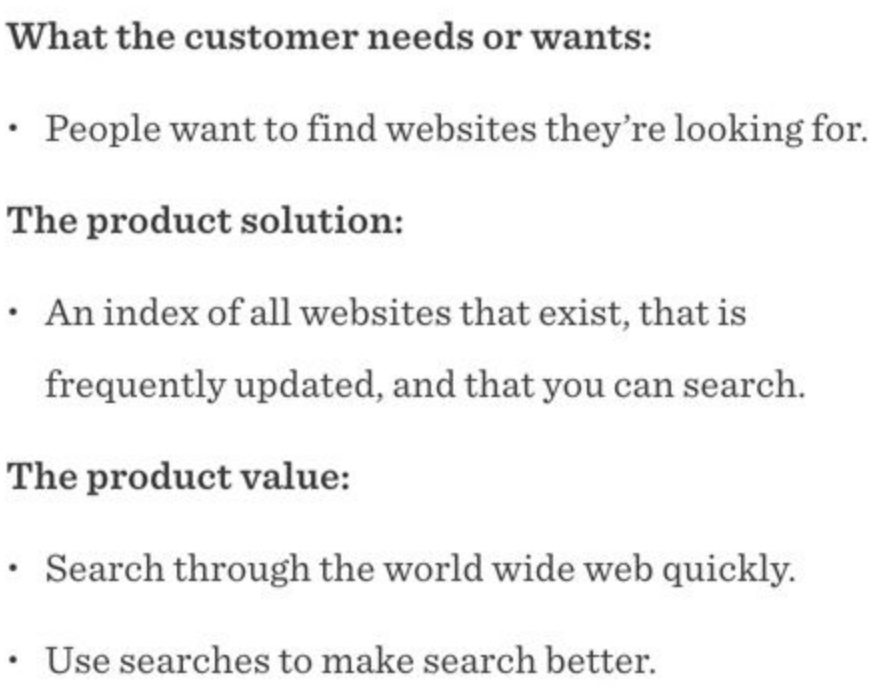
Building a product is problem solving. Instead of figuring out how fast a train from Philadelphia is traveling or how many green apples vs. red apples are in a bag, though, we're solving for how the product we're managing can meet customer/user needs and desires. So there are three key variables you need to define:

1. **The Customer's Needs/Wants** - the "what." This determines what your product will actually do/be, whether it allows the customer to do something that they wouldn't be able to do without the product, or whether it helps the customer do something in a way that's better or easier than they could otherwise.

2. **The Product Solution** - the "how." There're many different ways to satisfy the what. So how does your product do it in the way that's most compelling, most appealing to the customer? Which brings us to...

3. **Product Value** - the "why." You need to distinguish the benefits/features of your product.When there are multiple ways to solve a user's problem, knowing your product value will help determine what the best way is, and what you move forward with.

Once you have all those, you have an idea of the product you need to build. Here's an example of how this can look:



The more you practice this - no we're not screencapping the answers to the other questions - the better you're going to get at this kind of strategic thinking.

Product Value

- The "why"
- There are multiple ways to solve for needs
- What are the benefits of the feature or product you want to build?
- Benefits can help you pick which way (feature) you choose to solve for a customer's needs

Google In 1998 Or Lycos In 1995

Need

- I want to find websites that are relevant to me.

Feature:

- An index of websites that is frequently updated, that's access by searching it.

Benefit(s):

- Search through the world wide web quickly.
- Use searches to make search better.

Uber, Lyft, Or Any E-Hail Taxi App

What the customer needs or wants:

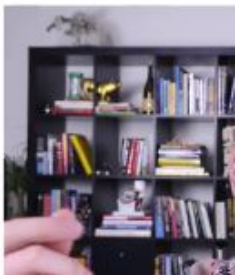
- Guarantee of a cab within minutes of needed.

The product solution:

- A smartphone app that lets you hail cars, and tells you when they're arriving.

The product value:

- Convenience, you don't need to be on the street
- Promise of a guarantee.



AOL Instant Messenger, SMS, Whatsapp, etc.

What the customer needs or wants:

- People need a way to send a message directly to one or more of their friends.

The product solution:

- A software application that lets you send or receive messages from many friends instantly.

The product value:

- Instant communication with your social network.

8. Soylent Green Is Customers! (How to Identify Customers)

In this lesson, we're talking all about people. Specifically, how you can identify the people who will most appreciate your product, and how you can talk to customers to

help develop a product that is even better for them. And we're doing it with more games, too!

You're building a product so that it can be used and enjoyed by customers - so it's important to identify who your potential customers are. Spoiler alert: they're people.

But if you don't know what people you're building your product for - what they like, what would help them have the best experience with your product - you're not setting yourself up for success. In order to find out about your customers, you have to **talk to them**.

Most product managers will be brought onto existing products with existing customers, but even completely new products (should) have customers in mind, or customers using a competitor's product.

Now, regardless of whether you're working on an existing or new product, you'll want to do some work on **customer discovery**. This is the process of identifying who are the right people for your product - who has the needs that your product addresses. Sometimes, an existing customer-base isn't the right one for a product. Sometimes there are whole new demographics that would appreciate your product.

In order to find that out, you need to go out and talk to your customers or prospective customers. Find out if what your product does/the problem it solves is important to them. Go out, get to know them, check out their collections of Back To The Future memorabilia.

Customer discovery is about finding who is the right customer for your product. The existing customer, isn't always the right customer for your product.

Is The Problem You're Solving Important To Your Customers?

- Go out and talk to your customers
- Talk to your customers, get to know them

Customers For Uber Before Launch

Existing Customers

- People who take yellow cabs in NYC
- People who travel a lot for business

Location of Customers

- NYC streets
- Airport
- Business office

Airbnb

Existing Customers

- People who stay at unique hotels (bed and breakfasts, castles, etc...)
- Motels
- Hostels
- Expert travelers (globetrotters, jetsetters, etc...)

Location of Customers

- Motels
- Hotels
- Online travel communities

9. Talk Consumer Survey To Me

Identifying your customers is just the first step. Now you have to talk to them and find out how best to address the want or need they have which your product solves. In this lesson, we discuss strategies for making the most and getting the best kind of information out of your customer interactions.

When you're talking to customers, you want to make sure you're always talking about the need or want that your product is (going to be) satisfying. Learning about how they use our/our competitors product will help us build a better product. But we **don't** want to actually name-drop our product. That's not the point. The point is effectively gathering the most information on how our product can be enhanced or changed or marketed or whatever else needs to happen to make it appealing to its customers.

There are several ways to do this, like **focus groups** and **customer surveys**. Both of these options are constructed very carefully and individually in order to extract relevant information about a product.

A simple framework for constructing effective information gathering is the **five Ws: Who, What, When, Where, and Why**. How does a product manager use them? Well, like this:

Who are your customers? For this, you can look at demographic data of your customer base. Is your product used by businesses or by individual consumers? In what industry or interest area? How old are they? Where do they live? And what's their income level? All these answers are going to help determine how your product addresses their needs.

What are their habits? What products do they use to solve their problems day-to-day, or guide them through their routine? What other products do they use? What are they already using as a part of their work? What do they already do on their phone or online? Understanding what people's habits are, and in different locations, will help you gauge the level of familiarity and utility your product offers them.

Where do they need your product? What is the environment in which they need your product? At work? At home? During a commute? In the middle of nowhere? What do they need to support your product, like a phone or a laptop? All these answers will help you think about how you need to build a product, what functionality it needs to have.

When do they need your product? Is it during a specific time of day, or at a specific event? It will help you build your product so that it's ready at the exact moment your customers need it.

Why do they need your product? This is the big, root question. Why can't they just use other products? Are there no other products that meet their need? Why did the specific need that they have, or the event that inspired their wanting the product, occur? Answering these questions are going to help you figure out how your customers think about their wants or needs, the terms on which your product can address those desires.

Those questions within the framework are all, of course, just examples. They can and should change based on your product, to give you the most detailed information with which to improve your product.

10. Not All Questions Are Created Equal (How to ask your customers questions - what are good and bad questions)

Now that you know who to ask and how to ask questions, we cover what sort of questions are going to be the most useful to you as a product manager. Regardless of

how well you've researched them and organized your focus group, the quality of the questions you ask will directly impact the quality of your product.

Avoid **binary** questions - they limit the amount of information you're going to get, and the more your customer talks, the more you'll learn.

You also need to avoid biasing **questions** - it doesn't provide you useful information, just information that is either too general or flat-out wrong.

Another bad question type are **hypotheticals** - they're asking people to predict the future, which isn't usually accurate, and can focus on your product rather than the problem your product needs to solve.

The good news is, with just a little tweaking, bad questions can become great ones. Like so:

Bad Question

- Have you heard of Airbnb?

Good Questions

- How do you find places to stay when you're on vacation?
- When you're planning a vacation, how do you figure out where you'll stay?

The focus shifts, from a particular service (Airbnb) to the user's experience - what their travel planning habits are, what tools they already have. Knowing that is going to get you the best information possible for building out your product.

10. Mock Interviews

As you work on your own interview scripts for customers, remember what happens here:

Starting with general questions will give you a fuller picture of the customer's habits and what matters to them, and this should absolutely inform your product.

Even the information that isn't specific to your product can be useful in planning new features to address customer needs.

Don't be afraid to tweak questions or adjust their ordering as the conversation changes and you have more information about each individual person.

11. No Cake For You: Risks And Assumptions

In this lesson, we clarify the difference between assumptions, which you need to make in order to build a product, and risk, which you need to minimize as much as possible. Plus, derivatives, dastardly figures, and different kinds of cake.

As you develop a product, it's important to understand that you're navigating risks and assumptions about it.

A risk is the possibility of a completely negative outcome.

An assumption is a possibility of an outcome which could nonetheless turn out in a different or unexpected way.

Whenever you build or manage a product, you're going to make assumptions - about how customers interact with your product, about what your product is going to be useful for. And by their very nature, these assumptions can be risks; if they aren't met, they could sink the product. It's important to identify the most risky assumptions underlying your product, and work to mitigate them.

12. MVPs: How To Not Throw Away Your Shot

In this lesson, we discuss MVPs - minimum viable products - and how they can provide market insight for your product without you having to do anything. Well, with you having to do comparatively less than actually launching a product.

Market: A group of buyers and sellers that exchange goods or services. Buyers determine the demand, while sellers determine the supply. A great/opportune market

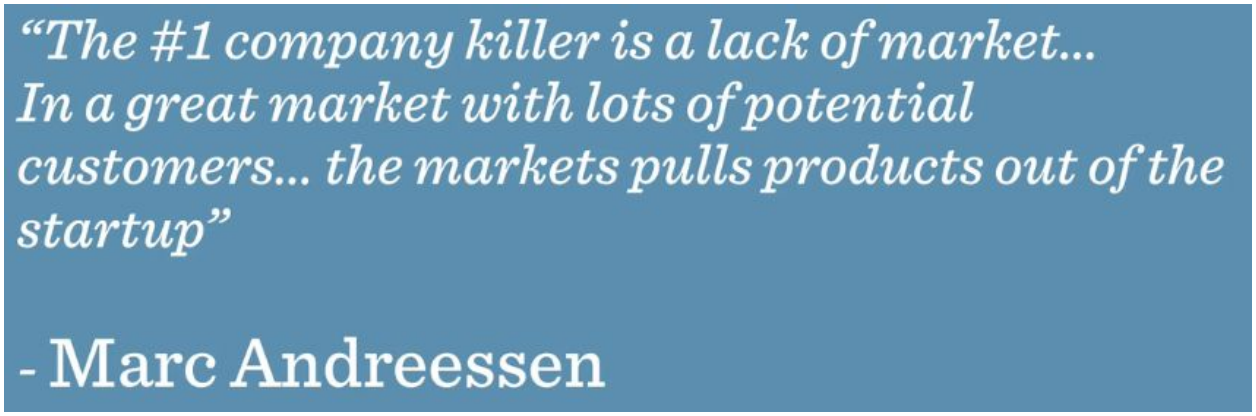
both has a problem that needs to be solved and enough buyer demand to sustain a business.

Product Market Fit: The determiner of a product's success: when it excels inside of a given market and found the "fit" between the buyers and sellers that makes it most profitable.

Minimum Viable Product: This is *not* a complete product. Your MVP is all about attracting customers, getting people to sign up and respond to your ideas, with a minimum amount of work on your end. You're looking to 1. validate or adjust your assumptions about your product and 2. learn the most you can about the market it's going to be a part of.

But MVPs don't mean you aren't doing any work at all. They are carefully crafted in order to test your riskiest assumptions (will people actually use this Dropbox thing?) and gauge the market demand (how many people are interested in a UX class from OneMonth?).

MVPs provide: Fast validation of your assumptions, risk reduction, feedback, all with low overhead (cost to produce it) and measurable progress of what you can reasonably expect of your product when you actually launch it.



13. MVPs: Many (and) Varied Persuasions

Now that we know what minimum viable products are, it's time to explore the types that you can build and what they test for. There are many different kinds but which one is yours?

Landing Page: Offers a pitch of what a product is; also offers the opportunity to drive traffic (create awareness) towards a site and get signups from interested users. It's a clear way to gauge interest in a product, but not much else.

Crowdfunding: Allows for more of a narrative/background for a product. There's (at least) a landing page and a video. Perks: people have to demonstrate interest by actually paying money, which reduces the financial risk of a product. But there's a potential to increase the product/technical risk: if you're funded, you're locked into building the product and/or pissing off a lot of people. Kickstarter isn't just a test. You're on the hook for delivery.

Product Demo: It showcases what the product will do - can look and feel real with little or no technical development. It also demonstrates interest, but little else. But the advantage here is that a product demo can clarify and demonstrate hard or integrated concepts.

Concierge: Delivers services manually to the customer, with the same steps and value as the full product, but without the risk of putting the product out on its own. Think about all the things you've found on GroupOn you wouldn't have known about otherwise. Those services use the security of an established platform to get themselves out there. The downside is that concierge demonstrates the value of a solution, but not the experience.

Wizard Of Oz: Or, wait, was Zappos.com not real? This is when the functionality of a product (specifically a website) looks automatic on the user's end, but is managed manually on the product end. This demonstrates the value and experience of a product, but doesn't actually provide the ability to deliver at scale.

Piecemeal: Similar to a Wizard of Oz MVP, it takes components that already exist in order to illustrate the value and experience of a product without actually building it out to scale.

14. The Magic of Making MVPs

Ready to build an MVP, nerds? We go over the steps of MVP development and go in depth on the first three: identifying risk and assumptions, converting those into testable hypotheses, and deciding which hypotheses to test.

Building An Mvp For Nerd

Customer's Needs

- People need random (nerdy) jobs to be done at a competitive price
- Stanford students are smart, have a great brand, and want to earn money

Product Solution

- Hire Stanford students for tutoring, projects, surveys, etc, with a simple text message

Steps To Build An MVP

1. Declare Risks And Assumptions - Well, okay. For Nerd:

Risks And Assumptions:

- People have random (nerdy) jobs they want Stanford students to do.
- Stanford students will want to do these jobs
- We can understand and price the jobs people send us (quickly)
- People will pay for these services
- We can price jobs at a profit, and so they're attactive to Stanford student.

2. Turn Risks And Assumptions Into Hypotheses - For instance,

Assumption

- People have random (nerdy) jobs they want Stanford students to do

Hypothesis

- People will be interested in Stanford students doing their nerdy jobs

The interest that people have in the service will prove the assumption that they have the need our product wants to address. As your converting, think about the actionable steps like this which are able to be proved or disproved.

3. Decide Which Hypotheses To Test - How many you test is up to you, but make sure you the ones that are existential to your product: the things that need to be true for the product to be worth building.

(Now that we know what assumptions we need to validate, we need to pick the MVP that will best test them.

As we decide what MVP we want to build, it's important to identify the kind of information we want to get from the user: do we just need to test interest, or whether people will actually spend the money? Do we want to be locked into building this version of the product, or do we just want to simulate its functionality so that users see its value?)

5. Define What Results Validate Your Hypotheses - Actually set the bar for what's going to be tested. How many requests are enough to demonstrate successful interest? How fast do the response times have to be? How many requests actually end in a transaction? How many of those transactions are profitable? Your job as a product manager is to make these calls.

6. Build An MVP That Tests These Hypotheses - This is the legwork. Just like on a normal product, it involves finding or creating the resources (students willing to take part, a landing page, a group text number, a Venmo account) that will allow the MVP to exist.

7. Launch The MVP - Tell your friends! Tell your family! Figure out how to drive enough traffic to the MVP to make the experiment viable.

8. Pivot, Learn, Iterate - After giving your MVP some time, analyze the results. Figure out what worked and what didn't; learn how to solve the problems that held it back; and build on the successes by iterating a version of the real product that makes the experience more seamless.

You can see what this really looks like: behold, NERD.

15. The Metrics System

Now that we know how to make an MVP, we need to learn how to analyze one.

Metrics are a unit of measurement of a product's progress and success.

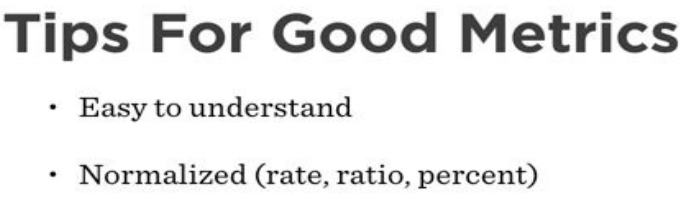
There are, as you can imagine, many different kinds.

One class of metrics are **Key Performance Indicators**(KPIs): **Stuff like sales, revenue, or customers.**

But metrics don't have to be that global or high-level. They can be related to an individual product page.

Metrics can be hard numbers, percentages, conversion rates, all kinds of things. But useful metrics can help you identify places you can improve your product.

Metrics provide objectivity and focus. It helps you sort all the feedback you take in and figure out what's meaningful, what's true, what you need to direct your energy towards.



(Isolated means - non dependent metrics, though this is less possible - don't worry too much.)

16. A Casebook Of Metrics

Haven't gotten enough metrics yet? Good! Neither have we. In this lesson, we identify common metrics across the different stages of product development. Knowing these will help you define success and create solutions for your product.

There are four actions metrics are built to track:

Acquisition - The new customers you acquire for you product. Common metrics include:

- Account Registrations
- Application Installs
- Account Or Application Install Conversion Percentage
- Cost Per Acquisition

Engagement - How your customers engage/interact with your product.

- Pages Per Visit
- Time On Site
- Event Completed

Revenue - The amount of money your product receives from customers.

- Time-Based Revenue
- Average Revenue Per User
- Average Time To First Transaction
- Percentage Of Paying Users
- Lifetime Revenue Per User

Retention - The amount of customers who continue to use your product.

- Retained Users
- Churned (Lost) Users
- Retention By Time-Base

It's great to be familiar with all of these examples. But remember that metrics are only useful as they help you identify places you can improve your product and approach your own measures of success. It's your job to determine what action is going to grow your product and how best to measure it.

17. Persona Of Interest

In this lesson, we get tactical and discuss how personas - or groupings of customers - help us organize products and make effective decisions. Pizza for everyone!

Personas - are simply groupings of people based on their needs or interests. These "types" are useful so that everyone working on the product can remember, relate to, and understand the product's customers.

Personas help us process all the information and feedback we get about a product more accurately. Because we as product managers are involved with the product, that familiarity inherently biases and warps our perception of it. Customers aren't in the room with you, or your friend group, or your area. They're potentially all over the world. Product managers have to think about these potential customers throughout every phase of product development.

So in Christian's pizza example:

We have to make choices about the product we're producing (pizza with different toppings) which cater to different personas (meat lovers, vegetarians, and cheese purists) we can identify, and yet please the largest number of people.

Remember: everyone is different, but people are similar. Using personas is a way to establish focus and make decisions with empathy, with an understanding of who your customers are and what they want.

We Listen To Our Customers Because

- We aren't our average customers
- We build products for our customers
- We need to represent all of our customers if we want to build a successful product

Here in Silicon Valley, we forget how skewed our population is, and we should frequently remind ourselves how abnormal we really are. The average person who uses a software based product around here isn't really very average.
- Alan Cooper

Personas represent the types of customers your product has, in a way that everyone working on the product can understand, remember, and relate to.

Personas help product development teams build empathy, establish focus, and make decisions.

18. Getting Real With Personas (Identify and share them - communicating customer needs)

In this lesson, we explore how to identify personas based on all your customer interactions, as well as common types of personas product managers should be mindful of. Spoiler: we're an Aries.

As we work to identify personas, it's important to keep in mind that of the information that goes into them comes from real people. A persona is a representation of a kind of customer, but it isn't fictional. The personas you identify should give you clear targets to aim for and allow you to build a product that will, in real life, meet the needs and goals of actual customers.

Steps To Building A Persona

1. Collect Customer Information - This comes through customer feedback or response to the MVP.
2. Analyze And Review Customer Information - This just means taking a close look at the feedback.
3. Identify Customer Similarities And Trends - For this, think back to the pizza example. This step involves grouping customers into like categories. You should take time to establish want connects all of these groups together.
4. Decide What Information Is Relevant - There might be a lot of similarities and a lot of trends which come out of your information. You need to decide what inputs are actually relevant to your product.
5. Create Personas - You may or may not want to name him George.

How To Present A Persona: Your persona should be easily legible to your whole team. So it's useful to have a quick take that breaks down the persona's skill set related to your product and immediate needs. You'll also do a gloss of the persona's key goals, habits and/or preferences, so that you have a sense of what that group of customers, represented by the persona, cares about.

Some examples:

EZ-Write primary persona: the main user type

Fred Fish: Corporate Chef

"Get me out of the office &into the kitchen."

Employer: Boise Controls
Background: Masters from Johnson & Wales University
Computer skills: Novice

Quick take on Fred	
Computer skills	**Novice** Expert
Job situation	**Employee** Director **Manager** Vice president
Computer type	**Netbook** Laptop Desktop
Computer tools	Advanced features Coding tools **Email** **Web browsing** **Word processing**
Background	Business Engineering Marketing Management **Other: cooking**

Photo from Flickr, some rights reserved

Key goals

Fred doesn't get his hands dirty the way he used to (literally). He stops in at all six Boise Controls sites as often as possible to stay in touch with cooks and cooking.

He wants to learn computer tools, but not at the expense of managing his kitchens.

A day in the life

Once a month, he meets with the head chefs and to plan the menu. When they're done, he sends it to his staff and his manager.

He's not a computer whiz. On a good day, he can drag in some clip art and do some formatting with fonts. Once in awhile, he'll format menus with the new editor they on his MacBrook Pro.

He's figured out a lot, but not everything. He always had a problem sending a message without the attachment, or an attachment with no message. That's annoying and embarrassing and used to keep him away from computers.

The new EZ-Write system seems to have some features to help with that. Anytime he writes something like "…see the attached menu..." the program prompts him if he doesn't attach something. If there were a Nobel Prize for software, he'd nominate the people who designed this.

Chefs are different from other users

Computers are just tools for a chef. Fred would rather use a cutting board than a keyboard.

Notice how all of these have different levels of information and focus, based on make sense for their different products. The level of detail you include in a persona can vary

based on the level of detail your product needs - the point is to get a personality across effectively to your team. The goal for you is to present a persona so that everyone understands who you're building a product for.

19. Persona Ex Machina

In this lesson, we cover the seven components that can help you create effective personas.

However you present your personas, the following information is good to have:

- **Name** - Great for ease of referral as you move through the process, so use distinctive but realistic names.
- **Descriptor** - Short, catchy phrases that describe a person's habits or their job title work best.
- **Quote**- This should capture the essence of who the persona is.
- **Who Is It?** - This is demographic information. Age, job information, location, skills and hobbies that are relevant to your product.
- **What Goals?** - These are their desires, their ambitions, their needs. They're also the opportunities for your product to interest them.
- **Which Behavior?** - What do they do, and how do they do it? Where among their obligations and habits does your product fit in?
- **What Attitude?** -How do they think about the product, service, or market that's relevant to you? Are they frustrated or excited about interacting with the product?

Whether you display all the information or not depends on what you need to convey to your team, but answering all of these questions will give you the fullest picture to, sing along if you know the words, build a better product.

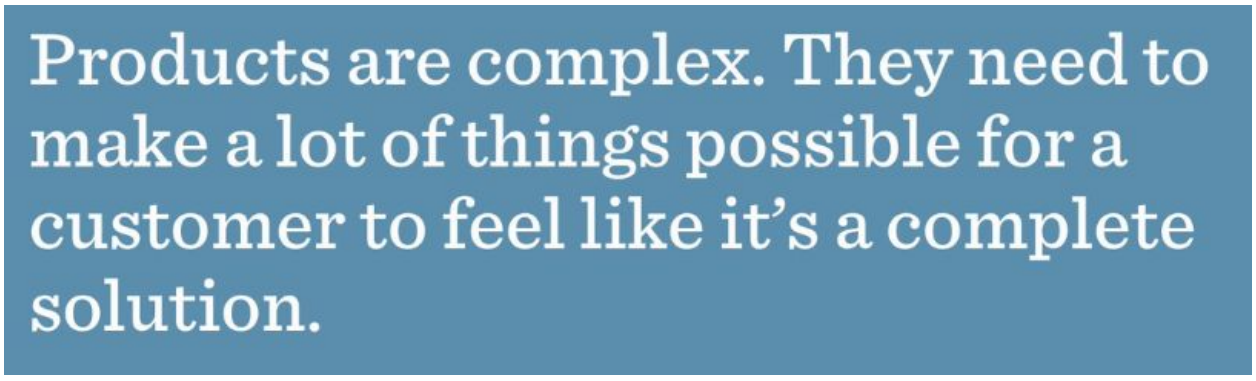
20. Features: Standing Alone Together

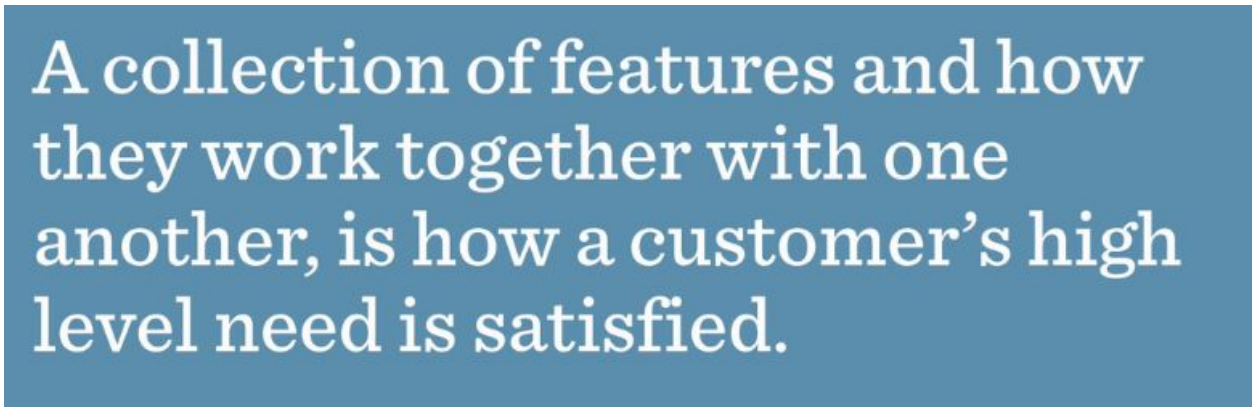
Now that we've covered the context and high level analysis of product management, we dive into actually building a product. The first step in this is understanding and identifying the features of a product that make it the complete experience you want for your customer.

Products are complex. They're made up of component parts that come together into a distinct, discrete entity. So in order to create something that successfully meets the customer's need, the actual process of building a product is one of assembling all those different pieces into a whole that can stand on its own.

For instance, a car is a product, but it's made out of many features. And each one, whether it's the anti-lock break system or the satellite radio, needs to satisfy specific requirements so that customer has a positive experience overall with the product. Or, put another way, while every feature satisfies a specific need, it is through features working together that the customer's high level need is satisfied and the product feels complete.

The next time you get on Twitter or Facebook, think about all the features that add to the user's experience and keep people using those sites.





As a quick recap:

Features- are functionality built into a product to satisfy a customer's specific need.

So, as we build a product, we need to identify those needs and create the features that satisfy them. In the topsy-turvey world where we are product managing Gmail, the **high level want** is a means for people to instantly send messages to other people on the internet. But there are a lot of factors and variation involved in sending internet messages, which different Gmail features have to address.

So based on customer research, you identify a specific need, and, even more specifically, how that need varies among different personas. Like, say, the need to send more than text over the internet.

A high-powered investor wants to also send spreadsheets to do with all her investing, but a popular jetsetter wants to share pictures of his trips with his friends.

These needs tell us that: people want to:

1. Send non-text information, which
2. Can be seen or worked on by
3. Accessing files through email.

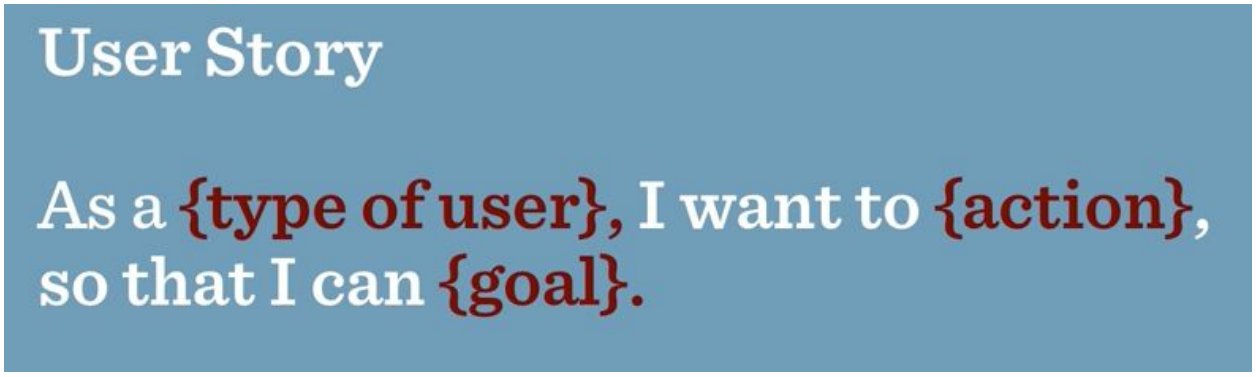
The feature that we build has to address all of these conditions in order to satisfy both personas' slightly different needs. How do we do that? By creating a feature that allows both these people to attach files to email.

That's the process of determining features - identifying customer needs, understanding the conditions that would meet those needs, and building a feature that offers a complete solution to those needs and satisfies the customer.

21. Features: Who Tells Your (User) Story?

In this lesson, we tell some stories. Specifically, the tale of how user stories can help clarify what features need to be accomplished and keep every team working on that feature on the same page, focused on the goals that will make the feature a success.

Basic common sense: lots of people are involved in the building of features and they all need to be on the same page about what they're doing and why. It's your job as the product manager to convey that information. An easy and consistent way to do that is the **user story.** They look like this:



A user story is a framework for articulating what a feature should accomplish, so that a specific customer can do what they want. They provide both focus and clarity to your teams, and even help define the feature. What's even better about user stories is their simplicity allows them to be used consistently across different personas and features.

How you make a user story is as simple as going into your personas and teasing out the three components - the **type of user**, the **action** they wish to perform, and the **goal** that action accomplishes. For instance, you look at needs of the jetsetter from the previous video:

Everything for his user story is right in the text:

As a popular jetsetter, I want to send more than just plain text, so that I can share photos.

But feel free to tweak the user story so that it reads cleanly and makes sense to your teams. It doesn't have to adhere word for word to your customer feedback/persona.

22. Prioritization Part 1: Some Assembly Required

In this lesson, we talk about how to organize all the pieces and needs we've identified and clarified, so that we can successfully assemble them into a useful feature. And all without an allen wrench.

The same principle applies for assembling a successful product and assembling an Ikea shelf without getting an ulcer: you follow the steps in order, each one building on top of the next, so that you can create a coherent whole. It's your job as a product manager to come up with that series of steps - and that process is called **prioritization**.

Since each distinct piece, or feature, addresses the users needs, then it's important at the start to prioritize those. In what order do the steps need to be built? Which ones are dependent on others?

Knowing when steps need to happen and what dependencies you have will help you organize your teams so they never reach Ikea-levels of frustration and so that you end up with the best possible product. It will allow you to structure the build so that you can estimate how long it's going to take, set a schedule, and plan accordingly.

23. Prioritization Part 2: Which One Of These Is Dependent On The Others?

Steps To Prioritization

1. Group Needs Based On Similarity Or Goal
2. Identify Group-Level Dependencies
3. Identify Group-Level Value And Effort
4. Prioritize Needs In Groups
5. Identify Dependencies
6. Identify Value And Effort

For Step 1, we can actually go through a list of our user stories and find which ones are similar. For our Gmail example, we'd group them together like this:

Now we're ready to tackle Step 2 - figuring what needs to happen first, second, third, and fourth so that we can satisfy all these groups of concerns. Which ones depend on others? So, here, it's probably a good idea to be able to send an email, since the rest of the needs are dependent on that.

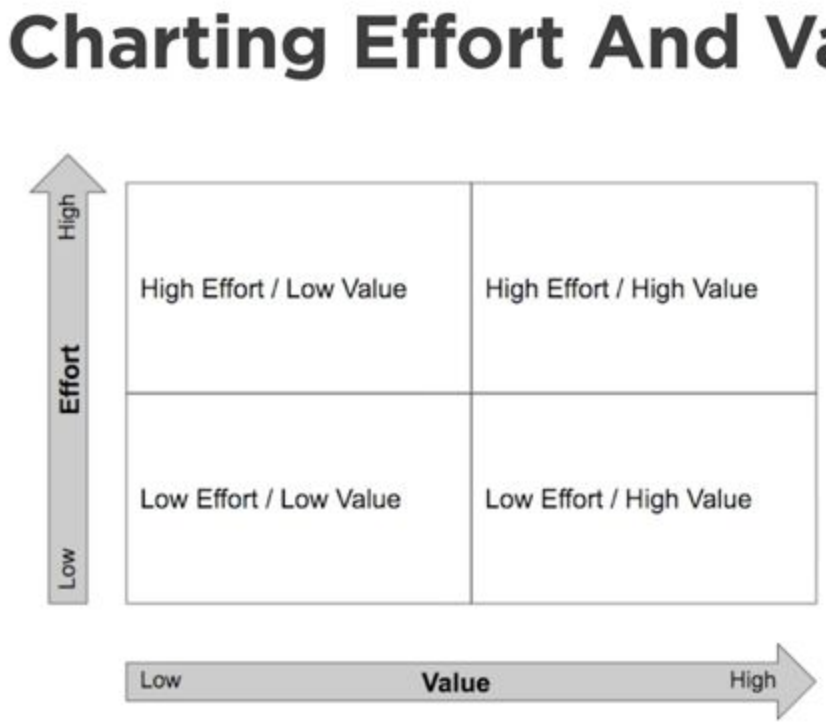
Value: the measure of importance, convenience, or indeed value a need is to a given user. How important is it to you to be able to send emails?

Effort: The measure of resources, time, and cost it takes to build a feature or product.

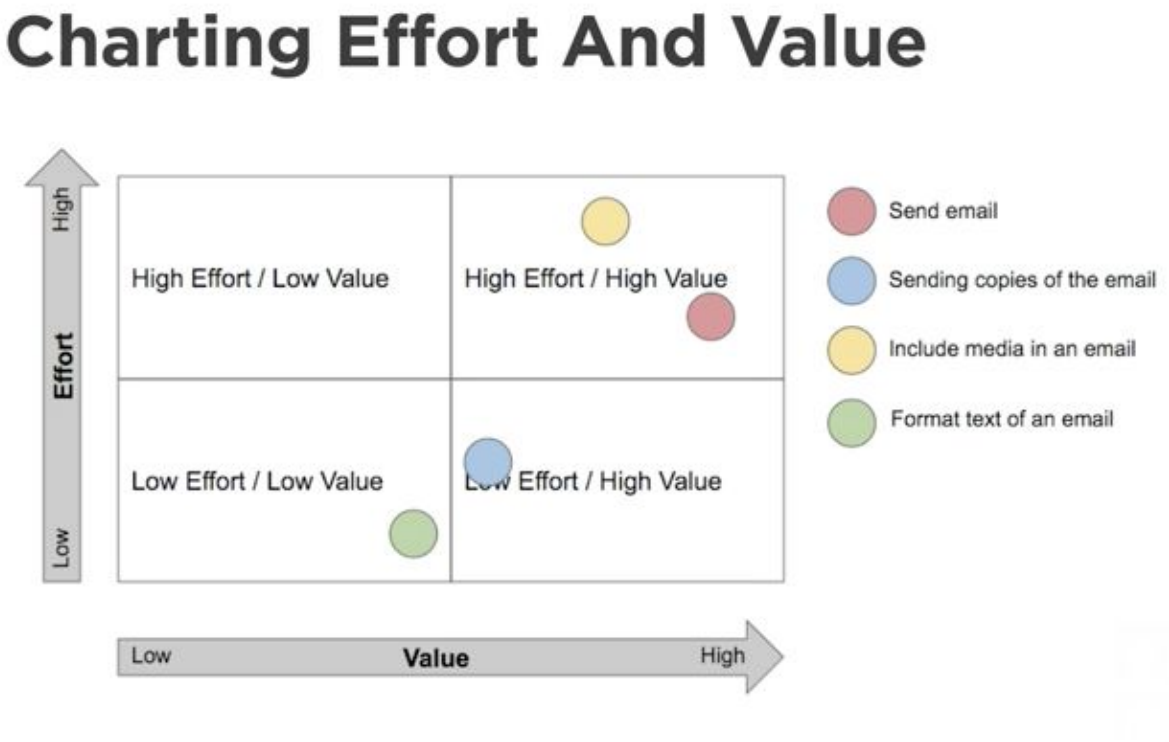
23. Prioritization Part 3: For The Greater Groups

In this lesson, we continue breaking down the effort and value of different features in order to prioritize what gets built first.

Where a task falls in your priorities is based in the intersection of its effort and value - the amount of work it takes to produce and how much it's worth to the user. All which can sound abstract and intimidating, so it's a great idea to break it down visually using a chart like this one:



Here's how value and effort break down for our Gmail example:



Based on what we know about these four tasks, then, we can decide what has to happen first, and plan our development teams schedules accordingly.

24. Prioritization Part 4: And Then There Were Fonts

We finish up our process of prioritization by talking through how to identify value and dependencies within groups.

The whole process of prioritization is to help you come up with the plan of action for actually implementing all those cool features and building out your product. Knowing what has to happen and in what order will allow you to develop a schedule, and keep both your teams and stake-holders informed.

25. Go With The User Flow

In this stage of the course, we talk user flows, a great tool for giving all your teams a consistent understanding of what they're building. They're also a great excuse for practicing your diamonds.

We've covered a lot so far, but it doesn't take pages and pages of notes to keep your team on the same page as they work. It can be as simple as a user flow.

A **user flow** is a diagram that uses geometric shapes to represent how the user interacts with your product. They're intuitive and easy to read, and can help your designer, engineer, or any other team member understand how a given feature works.

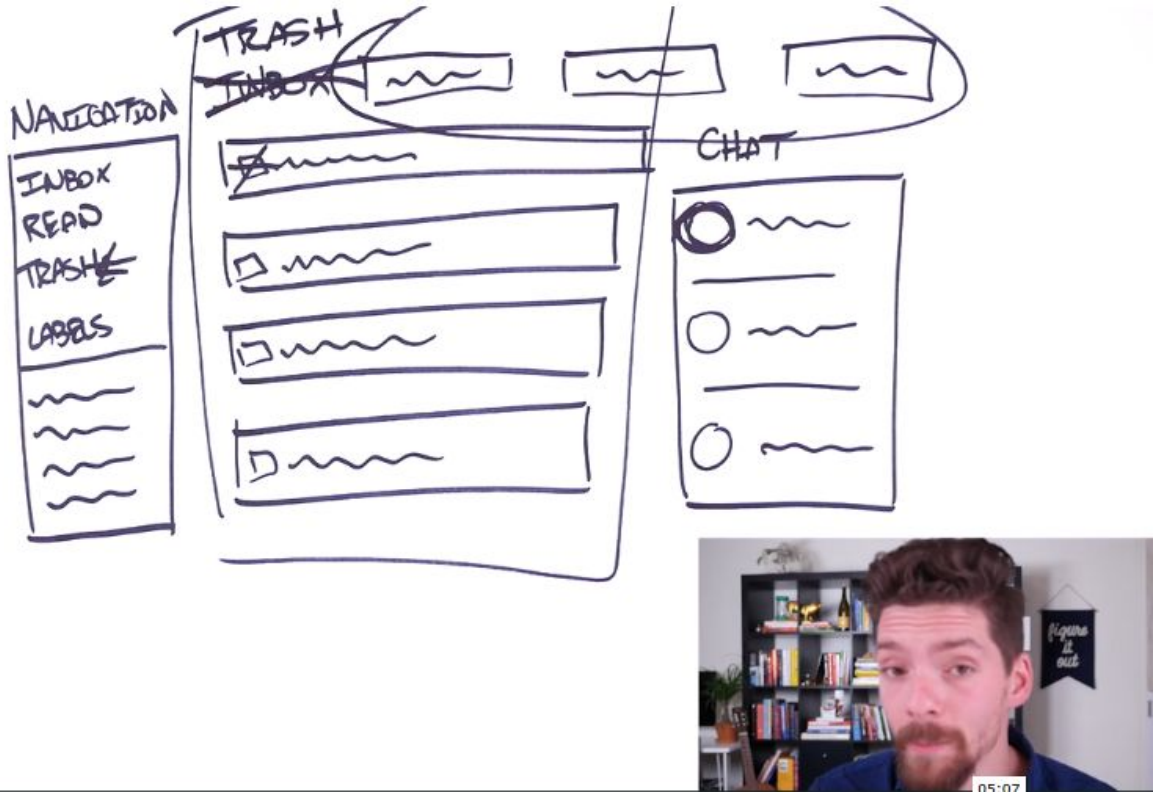
They look like this:

Being conscious of what a feature should actually do helps the people who build it make that happen. We promise.

26. Draw Me Like One Of Your Wireframes

In this lesson, we talk (and draw!) wireframes. As a product manager, you're going to have to work with both designers and engineers to help them understand what the product's actually going to look like.

Wireframes - Mock-up sketches which visualize what the product or feature actually looks like. Knowing how to make a good wireframe goes a long way towards realizing the product design that will best fit your users' needs.



27. Bring It All Together: The Product Overview

In this lesson, we talk about how we bring together all of the information we've gathered and all the solutions we've come up with. Product overviews will help you as the product manager create a concise, understandable way to access everything you've learned.

What We've Done So Far -

- Identified a problem to solve, or an opportunity
- Identified customers and their needs
- Identified a product solution and product value
- Researched our customers
- Launched a minimum viable product
- Formalized customers into personas
- Created user stories or jobs to be done
- Created user flows and wireframes

What We Have - A bunch of rich, valuable information about the product currently independent of each other.

What We Need To Do - Synthesize that information. We need to consolidate the product information into a central location and give a succinct answer to the what, why, and how of the product.

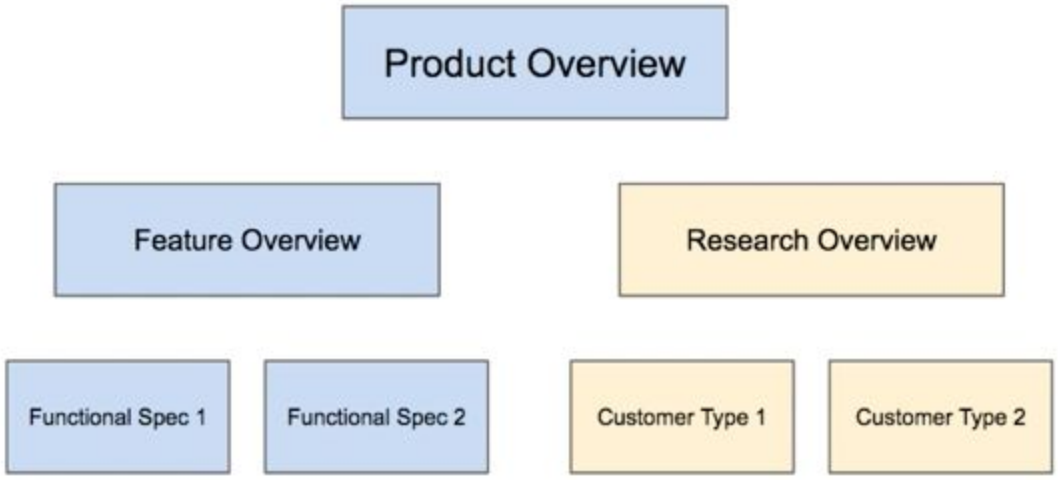
Product Documentation does this. It's a lot like a recipe - a series of steps that help you understand, plan and execute a product successfully with a team. All production documentation has an **overview** that answers three questions:

1. **What is the product we're building?** This includes a description, of course, but also identifies the customers, the goal of the product, and the assumptions, risks and dependencies of it as well. You can pull from much of the information you already have, everything from the original customer you need to identify to the analysis you pulled from the MVP.
2. **Why are we building it?** This includes a description of the problem or opportunity the product addresses, the hypothesis that's being tested. You should include the supporting data and metrics you've taken that gives you confidence of a positive outcome.
3. **How are we building it?** This section explains the features that solve specific customer needs, the priority of the features, and how the features actually work. For these, user stories, wireframes, and user flows will help illustrate your points.

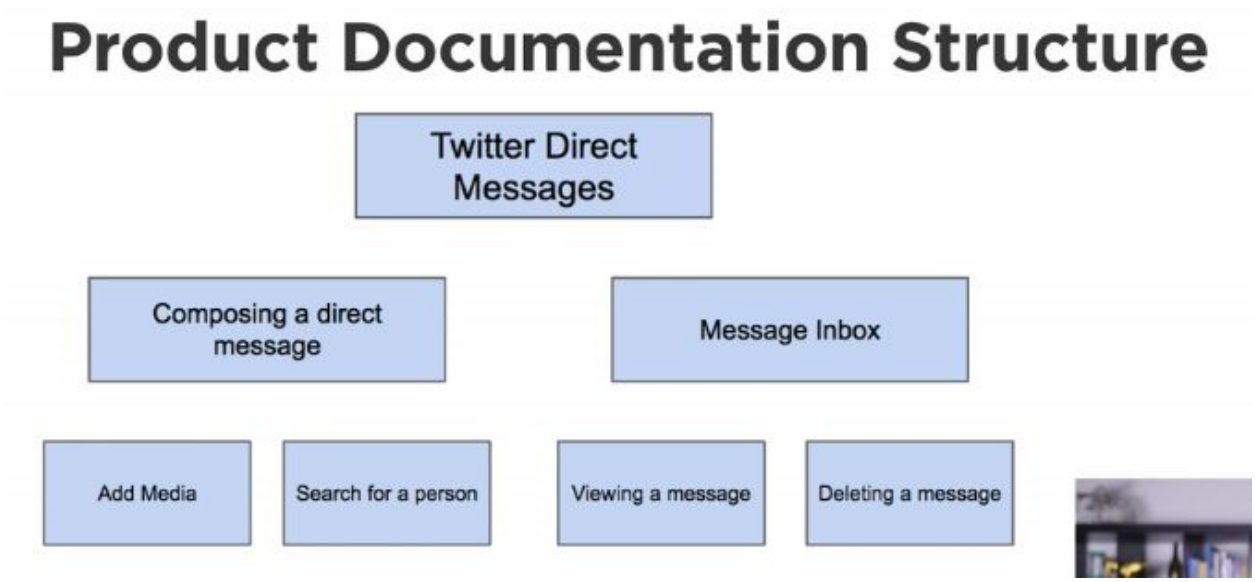
28 Papers, Please: Product Documentation

In this lesson, we talk about how to organize your product documentation beyond the overview so that anyone building the product or brought on later understand what the product's supposed to do and why it's built the way it is. It's like the Rosetta Stone but with more links to metrics.

Beyond the product overview, your product documentation should include the key features you're building, the needs they address, and how those needs are going to be addressed by specific functionalities. It can look like this:



It doesn't have to be a squat, multicolored pyramid like this one, though. The point of your product documentation is that you have something that packages all the information you have so that it exists in a form that's easy to share and understand.



29. Polishing Your (Product) Specs

Ready to go low-brow? Or at least low-structure. In this last lesson, we refine the best practices for creating specs, or the functional documentation, that defines product features and how specifically they work.

Functional documentation (specs) - the details of a feature at their most granular level.

Tips for good specs include:

- Use of wireframes, diagrams, and user flows to illustrate the user's needs and experience;
- Examples and scenarios to help someone understand how something works;
- Short, digestible blocks of text. Or, better yet, bullet points.
- Brief and to the point documentation
- Collaboration with your team to create the spec that works best for your feature/product

Specs should be immediately understandable, faster even than you're reading this sentence right now. They should be clear and easy to understand, and as visually friendly as possible. Consider these two examples:

Example 1:

> Direct messages can be managed by Twitter users. Managing a direct message gives the user the ability to view a message, mark a message as read, delete a message, turn off notifications for a message, and flag a message for spam.

Example 2:

Component	Description
Recipient	The twitter user to whom you're sending the message
Navigation	A link to get to your message inbox
Character Limit	A continuously updated character limit of your message
Send Message	A button to send a message; disabled if there's no content or if the user is over the character limit

One of these is easier for picking out relevant information than the other. Don't forget user flows and wireframes, either, which can get across the information you need even more visually. The goal is simple to illustrate, as simply as possible, how exactly a feature is going to work.

Example Of A Good Functional Spec

Twitter Users Can Manage Direct Messages

- Viewing a message
- Mark a message as read
- Delete a message
- Turn off notifications
- Flag a message as spam

Example Of A Good Functional Spec

Deleting A Message

- User selects to delete message
- Message is marked as 'hidden' in the database
- Message inbox only displays messages that are NOT hidden
- There is no way to view 'hidden' messages; for now

Example Of A Good Functional Spec

Wireframe For Deleting A Message

Example Of A Good Functional Spec

Deleting A Message

message_id	hidden
999	false
Action: User selects to delete message with message_id 999	
999	true
Display: message_id is no longer displayed because it has a value of true for the hidden field	

User Flow For Deleting A Message