Transcript of SRT Files

I'm currently a product manager at Uber where I'm leading rider experience for jump, bikes and scooters. Previously, I'd spent about eight years at Google and worked on a number of different projects, mostly focused on the intersection between hardware and software experiences. Some of these products you might have heard of, like Google Wifi, Google Home, and Chromecast. While I was at Google, I also spent some time at Google(x), Google Fiber and working on Ads. For school, I went to the University of Washington in Seattle, where I studied Informatics with a focus on Human-Computer Interaction. For those of you who haven't heard of informatics before, it's a combination of computer science, business, and design.

Product managers are often viewed as many CEOs driving the development, launch, and improvement for their futures and products end-to-end. At its core, product management is really about identifying and defining problems. In order to be successful, product managers must have a deep understanding of problems facing users. They need to understand user needs, they need to know their product, and usually the competition as well. In order to do this, product managers answer lots of questions. Product managers are responsible for figuring out what the team should build, specifically what the product should do, and it's not just about detailing what the product should do though, it's important to build products that people want. So the product needs to do something for the user or create value for the user. Understanding who your user is is also critical to building great products. Different types of users will have different needs. Once you know who your users are, it's important that you understand their needs on a very deep level. For example, imagine that you're building a calculator app. The way that you think about building it will be very different if your user is in grade school and just learning math skills. On the other hand, if you're building a calculator for a scientists working on pharmaceutical development, you would end up with a very different looking calculator. PMs also need to answer the question, why are we solving this problem? Why are we building this specific solution? It's important to understand the impact of solving the problem. Solving a problem needs to have both user value and also meet business objectives. It's also important to explain to the team why you're picking one solution over the others. Having compelling answers to these questions is what will move the project forward and ultimately, it's what we'll get people excited about working on it. Lastly, once you understand what, for who, and why, PMs need to answer the question, when, when are we building this? Unfortunately, you will never be able to build everything all at once. So it's important to understand and prioritize what to build first and what will go on the roadmap for later. There's always going to be tradeoffs to be made. These questions need to be answered in a strategic manner that tie back to business goals and objectives, as well as create a cohesive narrative around how these products fit into what the company's doing and what the company will be doing and can be doing in the future. Product managers are strategic. They're focused on building and bringing successful products to market. They need to be

hyperaware of their user needs, have intuition around how their users will respond to new features or changes in the product, and they need to be able to map out the steps that need to be taken in order to get the product to its full potential and vision. Along the way while they're doing this, they need to be able to course-correct as they learn new information. Product managers are constantly communicating out and responding to questions. They're the source of truth for the current state of their product; product shortcomings, opportunities, what the team's working on, and when those projects will be delivered. PMs need to communicate across a variety of different audiences, whether that's the design and engineering team, executives or customers. One of the most important communication skills is the ability to easily translate between different audiences. They need to be able to translate between user needs, to design solutions, to technical solutions, and more like business impact. Product managers also act as the glue that hold the company together, they're in the only role that sees the product all the way through the development process. Design might not be aware of what marketing is working on and vice versa, support might not be aware of new features that the engineering team spill. With this end-to-end visibility, product managers connect people across the team and share information in order to help facilitate reaching shared goals and objectives. Product managers also have to constantly respond to new information and decide what to do. Whether that's an email about a change in strategy from the CEO, a competitor launching a new product, or results from a user study. Product managers are always dealing with fires, deaf to constantly think on their feet and tackled issues as they come up. Like a production bug that prevents users from signing in or external dependencies that have slipped. So now you have an overview of what product managers do, let's move on and talk a little bit about why project management is important.

Now let's talk a little bit about why product management matters. Bringing products to market has gotten a lot more complicated over time. It's super easy to build products that fail, but it's really hard to build a great product. There's a huge need to have someone who's focused on understanding the user and the problem, so that the team does not lose sight of the important things that really matter. The product manager also plays an important role aligning all of the other teams that are working on the product so that they're moving together in the same direction, you'll get way better product outcomes when this happens. One of the most important things that product managers do is to identify and define what problems that teams should solve. It's not quite that simple though. Solving problems alone isn't enough, there's really an infinite number of problems out there that you could solve. It's important to make sure that you're solving the right problem. Not all problems that we're solving because they won't be successful products. Product managers make sure that the team build something that people want. This means you're solving a real problem for real users. Google Glass is a great example. The technology itself is amazing and quite impressive. Essentially, the team had miniaturized a cell phone into something so small that people can wear it on their face. But at the end of the day, Google Glass didn't really solve any problems for users. There was a lot of excitement around the product and some skepticism, but it wasn't clear why you

actually needed the product. You'll also want to make sure that the problems you're solving create value for users and the company, this means that your users will find enough value in your product to be willing to exchange some type of value. Two of the most common types of value exchange, are users paying directly for your product, are users being willing to see ads in your product. Another common pitfall here is sometimes teams create value for a niche group of users. A common way that this happens is that people will build things that they themselves want. Often, the product that gets built doesn't solve a need for enough people to create a viable business. Product management also plays an important role in connecting various functions in teams across the company. PMs advocate for the user, understand the business, empower design, creates stability for engineers, and they get everyone excited, but at the same time keep them focused and moving in the same direction together to build a product. Let's walk through a few examples of products with good and bad product management. Let's start with good product management. The first example is Whatsapp. Whatsapp started as a small company with an intense focus on being the best messaging app. They build a super easy to use communication app that many people use every single day to keep in touch with friends and family. Another example is OpenTable, which is a easy to use, hyper-focused app for making reservations at restaurants. It helps both users find a place to eat as well as restaurants to get people in their doors and streamline some of their operations through the use of technology. The last example is Headspace, an app for meditation with the aim of helping users manage stress, be more focused and sleep better at night. Headspace promotes daily meditation as well as ongoing use of the app. It's available to users through a subscription model. Now let's walk through a few examples of products with bad product management. The first example is MoviePass, which was a monthly subscription service that allowed subscribers to see one movie per day in a theater in the US. It had been around since 2011, but in 2017, they lowered their subscription price from $50 a month to $10 a month. A ton of people signed up. I personally use this service and it was great at first. I got a lot of value out of it. However, the company quickly started to find out that their existing business model was not sustainable. They started making constant changes to their product to reduce the number of movies that subscribers would see. Ultimately, they went out of business. Another example is everpix, which is an amazing product for managing and organizing photos through automatically sorting and pulling and finding the best pictures for you. This app was beautiful and it's very clear the team spent a ton of time perfecting it. However, they had a small user base and even fewer paying customers because they've not invested in growth and marketing or from differentiating their product from other freely available alternatives in the market. They also ended up going out of business. The last example is Google Wave, which was a product from Google that was supposed to be the future of collaboration. Imagine a product that combine Gmail, Calendar, Google Docs, Hangout and blogger, but that's still only half the product. The problem was that no one really knew exactly what this product was supposed to do, but everyone had super high expectations for it. The product had a ton of functionality to support a whole bunch of different collaboration use cases. But ultimately, that led to weigh too much complexity. One of the

ways that you can usually tell if a product is too complex is that if you can't explain what it does in a couple of sentences. Google Wave was also eventually shut down. So now you have an understanding of why product management matters and what can happen with good product management versus bad product management.

Let's talk a little bit about the history of product management. It's difficult to map out the entire history of how product management came into being. But, there are some key moments that we'll walk through. In 1931, Neil McElroy at Procter and Gamble wrote a memo called Brand Men in order to hire some additional people on his team. But it became a cornerstone for the evolution of product management. Brand Men were a part of the marketing team and they had full responsibility for managing the brand. They needed to complete a variety of task in order to do this, such as monitoring sales, advertising the product, and running promotions. Additionally, the memo also emphasized the importance of studying things personally at first-hand through field testing. This is an important part of product management today. In the 1930s, Bill Hewlett and David Packard met each other at Stanford. They also met McElroy as an advisor, and they were largely influenced by him. Together, Bill and David went on to found Hewlett Packard in 1939. One of HPs key strategies was to put decision-making as close to the customer as possible. By being closer to the customer, you have a better understanding of the problems that they're facing and their needs. Another moment was after World War II, when just-in-time manufacturing was becoming common in Japan. There was a large focus on reducing waste during the production process, but there were also two important principles. The first one is Kaizen, which means change for the better. This principle is that the business should be continuously improving through innovation and evolution. These small ongoing improvements can have big impact and benefits over time. Today, you see products that are constantly being improved and optimized. The second principle is Genchi Genbutsu. This literally means Real location, real thing. This is anchored in the belief that to truly understand, you need to go out and observe what is actually happening. PMs need to have a deep understanding of the problems that they're solving, and they use that understanding to help make decisions. Getting that understanding firsthand is a great strategy. It's also an eye-opening experience where you'll learn a lot and you'll also build empathy for your users. At this time, product managers were originally part of the marketing team and focused on the 4 P's of marketing. Making sure that the right product is at the right place, at the right price, with the right promotion, which is about explaining the value of your product to the consumer. But because most products had bad development timelines, the focus was less about the right product and more about the right place, price, and promotion. This role is known as brand management. Enter tech, as new technology products started to enter the market and create giant new industries, it was no longer possible to rely on focusing on just place, price, and promotion. It became critical that products were developed in order to meet customer needs. As agile methodologies became more common, software development teams became more collaborative across cross-functional teams, and product managers as a result, were able to focus more on the

user's needs and figuring out what the product should do. Depending on the company, place, price and promotion might still be a focus for product managers, but it's also common that these responsibilities are shared with marketing. We just quickly walked through a brief history of product management. Next up, let's talk a little bit about what to expect from this course.

Now I'm going to give you an overview of this course. After this lesson, this course has four more lessons. The first lesson is role of a product manager, the second lesson is about finding the right problem, the third lesson is about vision and strategy, and the fourth lesson is about communication skills. Let's talk a little bit more about each lesson. As I mentioned, the first lesson is about the role of a product manager. This lesson is focused on what PMs do, who they work with, and artifacts that PMs create like roadmaps and PRDs, which are also known as product requirement docs. The second lesson is about finding the right problem. We'll dig into how to identify the right opportunities, better understand the market, the user, and the problem, as well as create and test hypotheses. For the third lesson, we'll be focused on vision and strategy. We'll go more in depth on vision and strategy, and we'll also talk about the importance of MVPs or minimum viable product as well as different business models. The fourth and last lesson is about communication skills. We'll talk more about effective communication skills, how you can use storytelling to establish powerful visions, and lastly, how to build compelling, easy-to-follow presentations. Now that you have a high level understanding of the topics that we'll cover in the remainder of this course, let's walk through the learning objectives. At the end of this course, you will be able to describe the role of the PM and the PM's critical partners, build roadmaps and write PRDs, identify and size opportunities, create a business case for a new product opportunity, define the use cases of a product, including its KPIs, build a projected revenue model for your new product, and lastly, put together and present a compelling product pitch in order to gain internal stakeholder buy-in. So now you have an understanding of what to expect from the remaining lessons in this course. I can't wait to get started.

In this lesson, we'll go more in-depth on the role of a PM. We'll cover the following topics: the role of the PM; what exactly a PM does; who PMs work with; how PMs go about identifying requirements; and artifacts that PMs create to keep the team on the same page about what's going on, like roadmaps and PRDs.

Let's talk a little bit about the role of the product manager. Everyone has an idea of what a PM does, but there's some variability, and we'll discuss that. The one thing though that is always true is that at the end of the day, the role of the product manager is to make sure that the team is solving the right problems and successfully building products that people want. In order to achieve this, product managers are positioned in the center of design, business, and technology. On the design side, PMs understand the user's needs and their motivations. PMs are advocates for the user. On the business side, PMs understand the business goals, and

can align the product to help meet them. On the technology side, PMs understand how the product is built and can have technical conversations to understand some of the complexities, risks, and tradeoffs. It's common that PMs might skew more towards one-dimension or another, but at the end of the day, PMs need to have the background from all three of these areas in order to be successful. Product managers often act as many CEOs for their product when taking them from an idea to lunch. This is somewhat true, as they are responsible for the outcomes of their product. But a common misconception is that people often think that PMs are in a position to make all decisions. That's not quite the case. In reality, PMs rarely have an indirect authority over the team. Meaning that PMs don't manage anyone working on the team that's building the product. So it's important to be able to influence without authority and guide the team to the outcomes that you want. Instead, PMs work with and align everybody to define what the product should do and why it should be built. Building strong relationships is key. One of the ways I've found to build strong relationships is by getting to know people on the team better. I'll frequently ask people on my team what they think we should be focusing on, what they're excited about working on, or what they're worried about. Having this understanding of how the team is feeling is helpful in figuring out how to position and get the team excited about the things that we're working on. A large focus is on identifying and defining problems to make sure that the team is solving meaningful things. Before the team start solving a problem, they need to be sure that it's the right problem to be solving in this moment. PMs can provide this context by clearly defining and articulating the problem, why the problem matters and should be solved, and the outcome for what will happen if the problem is solved or not solved. Problems can matter for a lot of different reasons. Let's walk through a couple of examples of why solving a problem might matter. First of all, solving a problem might make your product more appealing to users. It might increase sales, potentially, it could reduce user churn, and in some cases, solving problems might be required in order to stay competitive in the market, or they might make your team more efficient, or they might be legally required if you want to continue selling the product. Generally, factors to consider are impact of solving the problem and the amount of effort to solve the problem. Focusing on the outcome that's needed is what will help to answer why the problem matters. It's also this that can get the team super excited. PMs also prioritize what the team is focusing on. It can be incredibly challenging to figure out what problems the team should and shouldn't solve. But no company has unlimited resources. So prioritization also becomes super important because the team cannot be doing everything all at once. You have to be comfortable with tuning out some problems at least until a later date. I found it's super helpful to think of prioritization is answering the question, what should the team be working on right now versus at a later time? Using a stack ranked lists can also be helpful as they force a lot of great discussion around prioritization. Here's an example. Oftentimes you'll end up with many important high priority P0s, and it can be challenging for the team to know which P0 is the most important out of the three, and what's the one that they should be focusing on first. By stack ranking, it's much clearer how important things are relative to other pieces of work. In this case, there's still three P0s, but it's clear that feature B is the most

important, and it's the thing that the team should work on first. In this example, this is just an arbitrary choice that I made, but in the real world, you'd use data in order to determine the stack rank of all these items. PMs are the spokesperson for their product. They should be able to answer almost any question about the product or know exactly where to get the answer. PMs should keep the larger team aware of plans, status, and roadblocks that the team's facing. Also important is the ability to persuade and secure buy-in from a variety of cross-functional teams and stakeholders. Remember, since PMs don't typically manage engineers or designers, PMs have to convince the team what they should be working on. PMs can do this by clearly articulating the problem, why it matters, and impact. These things are tied to a goal. They could be measured in terms of engagement, revenue, reduction in expenses, new users or apps for rating, just to name a few. But you want to get the team excited about solving the problem. The best PMs can lead a group through a conversation and have the group reach the conclusion that the PM was intending all along from the beginning. Once there's alignment, PMs are responsible for coordinating the development of a product across all relevant parties like design, engineering, marketing, and support. PMs need to understand all the functions that are required to build a product, and bringing in the right people at the right time. One more thing about being a PM, no day is the same and no day is predictable. There will always be things and fires that come up and need your attention. It keeps things exciting though. Some days you might be out of the office meeting and interviewing users, other days you might be going through design spring, or working through a technical problem with the engineering team, or watching users going through a usability study, or maybe going through quarterly business review, or meeting with the support team to understand top issues, or getting an e-mail from leadership asking why certain feature is behind schedule and why the team isn't working on another feature, or giving a presentation at a launch event and meeting with the press for interviews, or dealing with production issues, like when a server goes down or an iOS update breaks part of your app.

The role of a product manager can vary a lot based on the company, and the product, and what the team needs. Company size can have a big impact on the day-to-day role of a PM. At smaller companies, it's likely that you'll have a broader focus and more end-to-end responsibility. For example, at an e-commerce startup, you might own the entire app, including search and ranking, viewing our products page, and the checkout experience. Since the company is smaller and has fewer people, you'll have less support in place. This just means that you need to wear more hats. For example, at that e-commerce startup, you might be more involved with user acquisition, promotions, and analyzing the results. There also might not be as many processes defined. This means that you can sometimes move faster, although it can be a little bit chaotic. These types of environments, it's helpful when PM's can create some process and stability for the team. At larger companies, you will likely be more focused, and because of scale, you can go very very deep in a specific area. At a large e-commerce company, for example, you might focus just on ranking. There would be a

separate PM focused on search layout, another one focused on viewing product pages, and probably a couple more on the checkout experience itself. You're also more likely to have larger supporting teams to help you out, like research, marketing, and finance. This means that you need to help coordinate work across all those teams. Lastly, there's more process. This exists in order to make sure that things being worked on across the company connect, and that one project won't have a negative impact on another team like a change in the ranking algorithm that reduces the number of users that go to checkout. In large companies, it's super important to connect with other PMs to understand what's on their roadmaps and any shared problems that you may be able to solve together, or dependencies that you have on other teams. Product philosophy can also impact the role of a PM. In organizations that are product-driven, the focuses on solving problems for users. Here, product managers mainly focus on understanding, identifying, and writing requirements to handoff to engineering. This largely happens through research and talking to users. Then once engineering has the requirements, they will go and they will build it. Some pros here with this philosophy is that it's a very customer centric approach. This means that the product is more likely to be something that consumers want. But there are some challenges as well. Sometimes engineering can feel a little bit disconnected from the user and lack empathy. This can create challenges because sometimes engineers might not see the full picture and the requirements might feel arbitrary. An example here might be Amazon Prime. Today shipping is a huge convenience that provides a ton of value to users and makes it even easier to continue to buy products on Amazon. In organizations that are engineering driven, the focus is on solving technical problems. Engineers will build things and then hand them off to product. Here in these types of organizations, the PM's role is really more about figuring out how to package the technology and bring it to market. There are some pros with this approach as well, it really promotes technical innovation. Another thing is that sometimes users don't always know what they want, or they have trouble thinking outside of the box. But again, as always, there are some cons associated where in these types of organizations, it's more possible that the product won't ultimately resonate with the customer. An example of a product like this would be Google Glass. The technology was pretty incredible, but at the end of the day, it wasn't something that people really wanted or needed. There is another model though, a hybrid partnership, and this is really the best of both worlds. PMs end up writing the requirements, but engineering is included in the process of identifying requirements. One example of this would be including engineering in market and user research. Then, engineering goes and builds it, but product managers are also included in engineering design conversations. This gives product managers more visibility into how the product is being built and some of the decisions and tradeoffs around specific envelope implementations.

Type of user can also impact the type of work that PMs do. Consumer facing products are focused on solving user problems and providing value to the user. There's often more focus on simple clean UX. Nowadays, having easy to use products is really table stakes for consumer facing products. Additionally, for consumer products, the user is also likely to be

the person who purchases the product. The team working on the product is able to experiment and update the product relatively fast. On the other hand, for enterprise products, the focus is more on solving business problems like workflows and process improvements, although there are some user problems mixed in there as well. PMs need to understand both business and end-user needs. Another key difference here is that the bulk of the value that the product creates is usually to the company that purchases the product. This is also another key differentiator. It's ultimately the company that decides to purchase the product, even though the employees will be using it. These types of environments the user and the customer are not the same, and this is something that product managers and enterprise space need to keep in mind. One last thing is that in a B2B environment, there are lots more problems for PMs to identify any requirements to manage. Generally, it also takes longer to iterate and update the product as businesses tend to be more change averse than consumers. If the systems that they're using to run their business change on a weekly or monthly basis, it's likely going to cause a huge loss of productivity for them. Product managers can also build specialization through working on specific types of products. Here are a few examples. For software, PMs tend to be more focused on digital experiences like apps and websites. They usually care a lot about user experience and will also need to understand roll-out cycles and processes like App Store review. PMs can also focus on hardware, and they will be more focused on components, capabilities, cost, and supply chain. Hardware PMs will also understand the build timelines, how the factory line is setup, and likely shipping certification and import implications as well. Another key difference here is that once a physical product is built, it can't be updated easily. This is very different than software products. But in the case of hardware products, it means that it's super important to get things right the first time. Another specialization is data. Here, PMs can focus on how data can be used to build and power products, potentially through ML. Data PMs get super deep into data and numbers, but not just to analyze the product. Instead, they focus on figuring out how products can be built on top of lots and lots of data. Think self-driving cars and Netflix recommendations. Growth is another area that product managers can focus on. Here, PMs will grow the product through features that drive adoption, either directly, like a refer a friend feature, or indirectly through features that would convince the user to get the product. Growth PMs focus on getting new users and have a deep understanding of their funnel optimization, and acquisition, and onboarding strategies. Another focus is internationalization. International PMs focus on bringing the product to new places. This means deeply understanding the differences in those markets, differences in user behavior and expectations, and adopting the product to fit. In addition, these PMs have experience internationalizing products. For example, adapting the UI to support right-to-left languages or creating systems that can customize content in an app based on location. Sometimes there's also additional requirements in specific markets that must be met before the product can be launched there. These are just a few examples of specializations that a product manager can have, there's lots of other ones as well.

We've talked a little bit about the core role of a product manager, but PMs will focus their attention on different things during different phases of the product development process. Let's walk through a few examples. Identifying problems is part of the core PM role and is one of the most important things that a PM does. PM spend a lot of time defining the problem for the team to solve. Essential to this is building a deep understanding of the user, their needs as well as the space the product will be in. There's a variety of ways to do this, ranging from using research, whether that's user research, market research, or competitive research, to leveraging existing product insights and data, or other inputs to create hypotheses and test them. We'll talk more about this in lesson 2. Once armed with an understanding of the problem space and opportunity, PMs will build strategies for how to solve the problem through the creation of their product. These usually include high level overview of the product in feature and what it does, as well as how it maps to goals, objectives, and KPIs. We'll talk more about this in lesson 3. Once a PM has identified a problem and identified most requirements, it's time to start planning for how to tackle the remaining work. This usually involves working closely with design and engineering to understand how much time is needed to both design and build the product. Program managers can help to drive this conversation. For larger products that will take a lot of time to ship, it can be helpful to break into smaller tasks or milestones. Usually a milestone will have related tasks that could potentially be launched to stand alone. Then you can go on and lay on additional functionality and milestones on top of that until you get to the fully developed product. It's also helpful during this phase to identify key points of contact across engineering and design that will be working with you as the PM. In addition to you, this usually means one designer, one iOS engineer, one Android engineer, and one backend engineer. But keep in mind that this can vary as it's highly specific to your project. During UX Design, it's super important to work closely with design while they're building out mocks and the spec for the user experience. Your role as a PM is to provide context that can help inform design decisions. You can also help to guide design decisions using your product intuition, but keep in mind that you might not always be correct. Whenever possible I always recommend that you run large changes in the user experience through usability testing. This will surface any issues that users might run into with your product and you'll be able to solve and address them before development starts. Usability testing isn't always possible though and it might not make sense for small changes which might be better suited for A/B testing if you're worried about them. Once mocks had been finalized and the spec it has been handed off to the engineering team, it's time for implementation to begin. I always directly ask my engineers if they understand all the requirements and if there's anything that needs more definition or there if there's anything that's unclear. These questions can lead to some very fruitful conversations and let you know if there are any gaps that you need to fill in. While the engineers are building the product, it can also be helpful to check in and get regular status updates. This will surface any roadblocks that the team is running into so that you can help to remove them. This will also give you a sense if the team is on track or if dates are slipping. You can partner with program manager to keep a pulse on how things are going. Once implementation is complete, it's time

to test and validate that everything works correctly. You'll want to work with the QA team to review the test plan which outlines different scenarios that should be tested, the steps to be taken, and the expected behavior. You need to make sure that everything is captured correctly and that the most important scenarios are tested. The QA team will then test the product and identify any issues. At this point, you should also test the product yourself and file any bugs you find. I personally think that it's super important for PM's to go through their own product and make sure that it meets their standards. Once testing is complete, you'll have a list of issues that need to be addressed. Once everything has been built and tested, it's time to launch the product. This could go a few different ways depending on what your shipping. For small launches, they might just go out silently. For large launches, there will likely be some type of marketing and PR. This could be as simple as an update on your website or a blog post about the event, to a more comprehensive launch event with press. You should work with your PR and marketing teams to see what makes the most sense. You should also write a review any material that's being written about the product at lunch. For launches that might have more risk, you'll likely want to have a phase roll out over a few days or weeks. This will give you a chance to review data and make sure that the launch doesn't have any negative impact on metrics. After the launch, especially for large launches, it's really important to review how things went. This should include a couple of different dimensions, like changes in key metrics, user sentiment, and how the team thinks the launch went. You'll want to make sure that you also capture the impact as well. This is a moment though where you can celebrate some of the wins that the team have. But you'll also want to make sure that you identify what you and the team learned and what you would do differently the next time. This is super helpful to make sure that mistakes aren't repeated. Keep in mind that you'll likely be working on multiple products at the same time, and those products will likely be at different phases.

There are also some things that PMs need to do all the time, we'll walk through a few of those now. Communication is another important thing that PMs do. PMs communicate out about their product, status, timelines, decisions, risks, and blockers. The best PMs make sure that the entire team is on the same page. This can be accomplished through a variety of different mediums like presentations and conversations but all PMs will write PRDs as well where they frame their problem and document requirements for their solution. You can test if a PM is doing a good job communicating by asking people on the team what they're building and why. If you ask five people that question and you get the same answer back from everyone, then the PM's doing a great job communicating. However, if you ask those five people and you get six different answers back, then the PM has more work to do. It's also important to establish inbound communication, meaning that if people have questions about our product or initiative, it's important that they know that you're the right person to ask. Lastly, another aspect of communication is that PMs need to spend time on understanding the team or, and company priorities. This is something that is especially true for larger companies as PMs will need to understand strategy and vision for the larger company as well

as how their product fits in with all the other products that the company is working on. Another thing that PMs need to do all the time is coordinating. PMs are responsible for coordinating the development and launch of their product across all the various cross-functional partners involved like design, engineering, marketing, legal, and support. This doesn't mean that PMs do all the work but they facilitate conversations and they help to remove blockers or things that might be slowing the team down. They also make sure that everything that needs to happen actually does get done. There's a saying, when the launch went smoothly, the team did an amazing job, but if there's a problem with the launch, then somehow the PM managed to screw it up. Another thing that PMs need to do all the time is keep the team happy. People are way more productive when they understand what they're working on and have a way to measure their progress and impact. People are also more productive when they're happy. So it's super important that you find ways to keep your team focused and happy. There's also a handful of other things that will come up that PMs will need to be able to do. Oftentimes, you might need to help out in different ways to get something unblocked or to get things across the finish line. This could be things like creating a simple mock, reviewing small code changes, analyzing experiment results, writing an article for the help center, and many other things. At the end of the day though, PMs do it all.

So who exactly do PMs work with? PMs work with everyone. PMs are in a super unique position that they have seen the product throughout the entire product development cycle. They're in tune with the challenges that the product is facing across different functions and there's no one else on the team that has this level of end-to-end visibility. As a result, PMs act as a central hub and they help the larger team to connect all the dots. Let's walk through a few groups that PMs partner with. Keep in mind though that not all companies have these functions and some companies might have functions that I don't mention. The first partner is user researchers. Their role is to help discover key user insights and behaviors, which helps you answer important questions about your assumptions, business model, and product. User researchers will also perform usability tests with users to make sure that your product is understood, solving a need, and can be used without any issues. In terms of interactions that you'll have with the research team, it'll be important to align on research priorities and questions that need to be answered. The research team will share their findings from the research that they've done with you and together, the two of you can also participate in research and see things firsthand. Designers are also a key partner. Their role is to figure out what the product should look like and create delightful interactions for your users. Designers also play an important role to make sure that we're focused on solving the right problems for users. Some of the key interactions that you'll have when working with designers are aligning on problem definition and scope, reviewing PRDs in mocks, and compromising on ideal design solutions versus technical limitations or timeline pressure. Engineering is another key partner. Their role to solve a whole bunch of tricky technical problems while they're building the product. They also work super hard to make sure that the product is always up and running and working. The types of interactions you'll have with engineers will be reviewing

requirements, PRDs, and mocks, having discussions around feasibility and timelines, as well as creating a plan for tackling technical debt that's built up over time. Technical program managers and program managers are another key partner. The role of a program manager is super critical. They make sure that everyone is productive and getting work done. They keep a team on a schedule and they have a pulse of what's going on and will flag risks and schedule slips. Interactions that you'll have with program managers are generally around prioritization exercises, sharing timelines and status updates, and reviewing the product roadmap together. Sometimes there's a lot of confusion between product management and program management. So let's walk through some of the differences. Product management is about understanding the user, the problem, and the market, and creating a roadmap of what problems to solve and what needs to be built. Program management, on the other hand, is about understanding the team, your organization, and how to get things done. Program managers focus on increasing efficiency, resourcing, and reducing risk through repeatable processes. Program managers execute against the roadmap that you built. There's also technical program management, which is similar but works much more closely with engineering and is involved in technical details of software development. Oftentimes, TPMs were previously software engineers, whereas program managers are more generalized. Sometimes on smaller teams at certain companies, you will need to do both product and program management.

So let's go back and talk about other teams that you'll partner with. QA is and another team that you'll work with. The QA team is one of the hardest working teams. They work super hard to make sure that the product works correctly in a ton of different scenarios. When they find a bug, they write incredibly detailed bug reports with all of the reproduction steps required. They'll also spend a lot of time on increasing their testing capabilities so that they can continue to hold a high-quality bar while the number of features in your product increases over time. Some common interactions that you might have with a QA team include: reviewing a PRD so that they understand how their product is supposed to work, reviewing the test plan and expected behaviors so that they know what scenarios are the most important to test and can identify when something is broken, and working with QA to flag and prioritize bugs where they find broken experiences. Another team you'll partner with his Data Science. The Data Science team understands all of the numbers. They go super deep into the data so that they can provide you with key insights. They also can help you to design and roll-out experiments so that you can better understand user behavior and optimize your product. Because they're so great with numbers, they can help you to quantify all sorts of impact. Some of the typical interactions that you might have with a Data Science team would be: aligning on data science priorities, reviewing PRD's together, and reviewing results from previous experiments. Marketing and PR also play a huge role in helping to make your product successful. They focus on making sure that everyone knows what your product is and what it does and why it matters. They also generally will be responsible for managing your web and social presences and will lead user acquisitions to help you get new users. The PR team is super focused on

organizing press events and making sure that the right story lands. Some of the interactions that you might have with the Marketing team would be: aligning on marketing priorities, discussing the product positioning, reviewing launch announcements and presentations, going through any pre-briefings for interviews, and helping to support issues when they come up during a press review. The Sales team is another key partner. Their role is really selling the product. They are the team that brings in all of the money. But it's more than just that. They also build deeper relationships with your customers and have great insight into customer sentiment and needs as well. Some of the interactions that you might have with a Sales team include reviewing the roadmap and upcoming features. This could also include discussing feedback from customers in terms of product shortcoming or challenges and also discussing features that would make it easier to sell the product or close deals with critical customers. Another team that plays a critical role in your product is the Support team. Their role is to help customers when they run into a problem and they're frustrated. The Support team can fix bad moments and turn them into magical experiences. They also track the types of support interactions that they have which can help you to better understand the tough issues that your users are encountering. They also spend a lot of time improving the support experience through process and tooling enhancement. Some of the interactions you might have with the Support team include:reviewing PRDs and roadmap so that the team can create the right training and documentation for support agents, discussing top customer issues as well as discussing supportability. For example, is it even possible to support your product? Have you invested in building the right tools for the Support team to be able to help your users? Legal and Privacy teams are also important partners. Their role is to really keep you from getting sued. They work hard in order to reduce risk while still making sure that the team can innovate. The Privacy team in particular also plays a large role in making sure that you're using your user's data in a responsible way. Some of the typical interactions that you might have include: reviewing roadmap and PRDs, reviewing flows and messaging to identify any legal or privacy requirements, and you'll also spend some time discussing what data you collect from users and how it's being used. These teams play an important role in flagging new requirements to you. The Internationalization team is also an important partner. Their role is to work with you closely to bring your product to new markets around the globe. A big part of that is translation, so that people can use your product in their own language. Some typical interactions include: reviewing expansion plans and prioritizing them, as well as reviewing any bugs that come up during the internationalization process. These are some of the most common teams that you'll partner with as a PM. But there are lots more. Keep in mind that it really does take a village to launch a product.

When you join a new team or company, it can take awhile to ramp up. To get started on a good path, here are the things that I would try to do during my first few weeks on the new team. I would want to spend some time learning about the company. Specifically, what the company does and what's the company's mission and what are the products it's building and why. I'd also want to better understand how does the company make money? Is through

selling our product directly to consumers? Is there an intermediary? Is it supported by ads? Are there key partnerships that bring in money? I'd also want to understand short-term goals and objectives. What are the short-term goals and objectives that the companies hoping to reach in the next three to 12 months? As well as, what are the longer-term goals and objectives? I'd also want to get a better understanding of what our current projects in flight? What are people currently working on and does it make sense for me to get involved in any of these? I'd also want to meet a whole bunch of people that I've listed below. But here are the main ones that I'll call out. I'd want to meet my manager and my manager's manager to better understand the team's goals, what I should be focusing on and their expectations for me. I'd also want to meet other PMs to understand what other people are working on. As well as research and design partners, in order to get connected to past research that's been conducted and better understand the design vision for the product. Equally important, would be connecting with engineering and program manager partners, to better understand some of the engineering processes in terms of how the product is developed and are there any areas where we have technical debt? I'd also want to spend some time getting a better understanding of the product experience. This means using the product and learning as much as I can about it. I do this by checking out the product's web page. How is the product marketed and positioned? Is it easy to understand what the product does and why I would want it? Can I figure out how I would actually get the product? I'd also want to review the app store listing for similar information. How is the product marketed and positioned? What category does the app fall into? What's its ranking? What's its rating? What do the review say? Do the posted screenshots show the most important aspects of the product? Then I'd want to use the product a ton. Go through every single flow and every single screen, learn how the product works and learn how to use it. During this process, I'd keep a journal of my experience using it. This is one of the most helpful things that you can do on day 1. You can provide meaningful feedback to the team. As part of this journal, I'd also include questions that I have about why it is the way that it is, as well as a list of issues that I encountered while using the product. While experiencing the product, I'd also want to experience what the support experiences like when I need to get help. This includes reviewing the support site, checking to see if it answers any of the questions I had or the problems that I ran into. But also I would want to reach out to the support team to see what the experience is like. What's the tone? Are agents helpful? How much time does it take to get my issue resolved? Lastly, I would want to try using competitor products to understand how other people in this space are approaching the problem, including comparing similarities and differences between our product and the competition. Lastly, there's a few other things I would want to do, including better understanding process for the team, specifically around how to get stuff done. How does planning work? How do launches work? How do roll-outs work? It's super critical to understand what are the types of things that need to be reviewed before the team can proceed, as well as what are the types of things that need to be approved before the team can proceed or launch something. Your first few weeks is also a great time to make sure that you have access to all the dashboards and tools that you need. Sometimes it can take awhile

to get all their permissions, so requesting early is a good thing. Lastly, I would want to shadow the support team and listen to customer calls. This can be super informative to hear firsthand from customers the types of issues that they're encountering and how the support team is able to solve them. There will be lots to learn, but even while you're ramping up, you can still provide value to the team by providing a fresh perspective on your experience using the product and documenting the issues that you run into. Make sure to ask lots of questions.

Once a problem has been identified as something we're solving, it's time to start identifying requirements for the solution. Identifying requirements involves answering questions about what the product should or should not do. This is also sometimes called requirements gathering, but it's much more active than just gathering requirements. The best PMs will understand the context behind each requirement. This is another critical element of a PMs role. PMs write requirements in PRDs which we'll talk more about later in a later concept. Identifying requirements can happen through a variety of channels. One method is through research. It's usually helpful to get started by doing some research, whether this is doing quick online research on your own to get a better understanding of the space or looking at information that the teams already collected, or doing more exhaustive market in user research, which can take many different forms, like side-by-side product comparisons with longitudinal space. Related to user researches is user interviews, where you can talk directly with users to understand their needs. This is a super powerful way to hear firsthand the types of problems or challenges users are encountering. It can also really help to build empathy so that you're able to put yourself in the user's shoes. Another method is through stakeholder interviews. Interview people at your company to get a better understanding of the business objectives and requirements that might come from other stakeholders like marketing or legal and privacy. You also can identify requirements through prototyping. By going through the process of building a prototype, it forces you to connect all the dots. You may realize that you're missing a screen or that there's a critical interaction missing that hadn't yet been accounted for. These are some of the most common methods to identify requirements, but keep in mind that this isn't an exhaustive list. There's lots more. I'm going to share a couple of tips that will be helpful while you're identifying requirements. Keep in mind that when you're talking to people, they might not always be able to tell you what they need, or just because someone says that they need something specific, it doesn't mean that that's the actual solution or requirement. The role of the PM is to push deeper and understand the motivations and needs that might not always be super apparent. For example, imagine after running some user research sessions for a budget managing app, you see that there's a trend of users saying that they wish the product allowed them to categorize each transaction. You could go ahead and build that and it would provide some value. But if you asked a few more questions, you'd realize that the user really wants to understand where money is coming from and how it's being spent. Instead of just adding a field that allows users to input this information, a better solution would be a feature that automatically categorizes each transaction for the user. Another tip is document everything. This includes both documenting

requirements as well as the steps that you took to get to those requirements. It's super helpful to have detailed notes that you can refer back to later on from researcher interviews that you did or meetings that you went to. Also, it's not just for you, it's for the rest of your team to refer to as well. One challenge that often comes up is that you can never be sure that you have all the requirements. Requirements will continue to grow and evolve over time as you learn more and get new information. When you're talking to people, you might not get all the information or they might forget to mention something important, or sometimes things change and priorities shift. Whatever it is, requirements will continue to grow and evolve over time as you learn more. This is okay, you can mitigate against this by factoring in and making sure that the team has some extra time to adjust to changes and requirements in the schedule. Another thing that ties into changing requirements that I cannot stress enough, is keep the PRD up-to-date. Things will inevitably change and as things do change, make sure that you update the PRD and include a change log with what changed and when. That document is the source of truth for everybody. Anyone should be able to look at it at anytime and understand exactly what the team is working on. Also, it's important to be transparent. Share out status updates and the requirements that you've captured and new requirements when they come up, and any requirements that might have changed. It's super helpful to make sure that the teams aware of everything and on the same page, but this also makes sure that there's alignment behind everything that the team is doing as well as making sure that your understanding and the teams understanding is the same. This will prevent many surprises from coming up in the future. These are just a few tips that you can keep in mind to make sure that you're setting yourself up for success.

What is a roadmap? Well, a traditional roadmap is used to get you from where you currently are to where you want to be. A product roadmap is used in a similar way. It's a high-level representation that visually shows your product strategy over time and where you're trying to get to. Roadmaps keep the broader team informed about priorities and upcoming projects. They're also super helpful in answering the question of how all the projects come together. When you communicate out roadmaps, they also become a tool to facilitate discussion with internal stakeholders to see if their priority and ordering is correct. What the tradeoffs are of changing the ordering or trying to accommodate an urgent feature request with the goal of creating alignment. Roadmaps work best when they tell a story about what the team is going to build and how everything is related, and why these projects are so important. You can do this by focusing on themes or goals and linking those specific projects back to those themes and goals along with an estimated impact. Share your roadmap out with the team and leadership. Roadmaps are great tools that will facilitate conversations asking if we're building the right things at the right time. They're also great tools because they can be used to clarify that in order to accommodate a new request, another project will need to be deprioritized. These types of conversations make sure that the team is all aligned and on the same page. Say no. One of the hardest parts about product management is deciding what not to build. There's an infinite number of features and improvements you can make to your product.

Likely you'll have lots of people asking you to build lots of things, but the team has a limited amount of time, so make sure that they're using it on the most impactful efforts. There's a great quote about this. Steve Jobs said that innovation is not about saying yes to everything, it's about saying no to all but the most crucial features. This type of mentality allows the team to focus on the things that really matter and execute with high-quality, instead of trying to do all the things and ending up with a mediocre product. Lastly, make sure to attach goals to items on your roadmap. This will help people to understand the impact of each project and answers the question of why this specific thing matters and is important. It also makes it easy to compare how one item compares to another item on the roadmap. Let's walk through a few example of roadmaps now. Here's an example of what a simple roadmap looks like. The first column shows the strategic theme or goal, then across the first row, you have time. Here I'm showing quarters, but it could be even more granular. Then I've mapped out all of the features and projects that the team will be working on throughout the course of the year. You can see that the team will be working on improving user on-boarding and entering new market in Q1. We expect this work to increase conversion by 10 percent and allow us to launch our product in three new markets. In Q2, the team will continue focusing on improving user on-boarding, as well as addressing iOS 13 compatibility issues. We expect feature A to improve conversion by three percent, and the work for iOS 13 will make sure that our apple continue to work with the new version of iOS as people update their phones. In Q3, the team will start the first phase of the migration project as well as addressing Android 10 compatibility updates. This will speed up our services and allow the engineering team to sunset the old service. The work for Android 10 will make sure that our app works well with new Android phones. In Q4, the team will complete the second phase of the migration project. Keep in mind that it's helpful that each item on the roadmap has its own PID, which can provide more details about the specific problem and requirements. Remember, this is just an example of a simple roadmap. You can add a lot more detail and attach even more specific goals and impact. You can also structure your roadmap by team. This can be helpful to see who's working on what and identifying when a team might be overloaded or have some extra bandwidth. This time, the first column shows what teams will be doing the work, and the first row continues to show time, same as in the last example. Let's take the same feature set and map it out by things that each team will be working on throughout the course of the year. You can see that the entire team will be working on future being Q1, and it will require coordination across back-end, Android, and iOS teams. The back-end team will also work on feature C, and the Android team will also work on feature D. In Q2, the back-end team will focus on feature A and the iOS team will address compatibility issues for iOS 13. In Q3, the back-end team will start the migration project and the Android team will address compatibility issues for Android 10. In Q4, the back-end team will focus on the second part of the migration project. You'll see that the back-end team is pretty busy throughout the entire year, but the Android team doesn't have any projects for Q2 and Q4, and iOS doesn't have any projects for Q3 or Q4. This type of layout will give you a sense if the team has some extra bandwidth so that you can squeeze in a few extra projects.

So now I'm going to walk through the roadmap that I put together for the alarm clock app. To do this, I'm going to imagine that the team will start delivering features in Q1, and I'll build out the roadmap for the rest of the year. Since an alarm clock app is pretty simple, I bucketed features into two themes, core functionality and extras. We're going to frontload the core functionality of the alarm clock app earlier on in the year and then focus on extras. So what this means is that in the first quarter, the team will work on delivering basic alarm functionality, which includes setting an alarm, turning it on/off, snoozing, and editing. Then in Q2, we'll focus on adding support for multiple alarms, including better management capabilities so that it's easy to add, edit, and remove multiple alarms. Then in the second half of the year, we'll shift and start to focus on extra functionality that will make the alarm app really stand out. In Q3, this includes a focus on the alarm tones, both support for custom tones as well as support for gentle alarms, which will gradually increase in volume over time. In the last quarter, we'll focus on having the alarm branch out to include automatically creating alarms based on calendar entries. Another thing that will be interesting to explore is sleeping and waking up go hand in hand. So the team is going to take a look at a feature related to sleep monitoring to see if we can understand the quality of the user's sleep as well as wake them up at the right moment in their sleep cycle. Remember, when you're putting together a roadmap, it's also important to include stakeholders like the engineering team to make sure that this is a reasonable timeline.

The PRD or product's requirements Doc, is the most important artifact that PMs create. PRDs align the team and make sure that everyone is moving in the same direction and solving the same problem. PRD is clearly spell out what is being built, who it's being built for? Why solving this problem is worthwhile? PRD serve to inform the rest of the team, whether that's design, engineering, or marketing, or support about the work that they will need to do. The best PRDs do a really good job explaining not just why things need to be built, but what are the right trade-offs to make? PRD serve as a living document and will continue to evolve while the team is working and iterating on the problem. The PRD should be updated when decisions are made or when the team gets new information. Here's an example of a simple PRD template. The first section is background. You'll want to frame the problem and provide any information that can help people to understand the problem better. This can also include past efforts. The next section is the problem. You want to describe the opportunity and why it's something that should be solved. What's the benefit to the user? Then in the goal section, you'll get more explicit about what successful look like. In the success metrics section, you define how you'll measure that goal. The next section is for describing features and scope. This is where you'll define the requirements for what the product should do. This is the most important section that design and engineering will reference. Lastly, I've found that it's super helpful to include mocks in the PRD, to make sure that everyone ends up on the same page of what's being discussed. Make sure to work with your designer on this. The PRD is super important. I've seen teams struggle when PRDs don't exist or when PRDs are incomplete or

not updated, you'll have a whole bunch of people trying to move the product in different directions. Oftentimes, this means that people might not be working towards the same goal. It doesn't work very well and it looks something like this. Whereas on the other hand, when you go through the process of writing a PRD, it aligns the team, it makes sure that everyone is on the same page and understands and is solving the same problem. Also make sure that everyone has the same goals. Then, the team will start to move in the same direction. It looks something more like this. Anytime the team is working on solving a product problem, there should always be a PRD, even if it's a simple one pager mini PRD. As the problem gets bigger and more complex, the PRD will also get longer to reflect that complexity. Most companies have their own templates for PRDs, which makes it easier to read, understand, and provide feedback on specific PRDs. But generally, all PRDs will do the following. They'll frame the problem and answer the question, why are we solving it? They'll outline the goals, both user goals, business goals, and success metrics and they'll describe the requirements. What does the product do? Remember, as a PM, you're answering what the product does. Design in engineering have to figure out, how the product does it. Additionally, there's also other components that can be helpful to include in PRDs, such as assumptions, things you think are true, but you're not quite sure, other options considered and why they didn't make sense. This can be really helpful for when people who aren't as familiar with the product, review the PRD. You also can label things as out of scope. This means things that you are explicitly deciding not to sell for and why. As I had mentioned, including UI mocks can also be helpful because it's often easier to communicate some ideas visually instead of through text. You also can include risks and mitigations. This section we'll talk about things that could go wrong and how you can prevent or minimize those bad things from happening. Lastly, PRDs can also include a support plan. What are the top issues users might run into and how will we help them? The exact format of a PRD will vary based on what company you're at. But PRDs always frame the problem, outline the goals, and explain the required features. Keep in mind that each item on the roadmap should have its own PRD.

Now I'm going to walk you through the PID that I wrote for an alarm clock app. The problem is users don't always wake up at the right time. They have a need for something to wake up or alert them when it's time to get up in the morning. Another challenge that users run into is that wake-up times might vary based on day of the week. For example, many users will wake up at a different time during the week compared to the weekend. For this specific example, I've also decided to focus the problem on alarms for waking up. Even though there are many other times when users might need alarms, this will allow us to start with a more focused feature set, and it won't prevent us from expanding to other use cases in the future. For goals, we want to make sure that users wake up on time. We'll measure this by the number of times alarms go off. The premise here is that if our product is not meeting our users goals, they'll stop using it. Potentially, we could also do something fancier like looking at if the phone is moved after 1, 2, 5, or 10 minutes after the time when an alarm goes off. We also want users to have the ability to set different alarms based on their schedule. We'll measure this by the

number of unique alarms created and the number of edits made to alarms. The premise is that users won't need to edit existing alarms. Instead they can create different alarms based on factors like day of week. Here are some of the features that I think will be most critical; the ability to have alarms based on day of week, support for at least 10 alarms, having the ability to manage your alarms. This means editing and deleting existing alarms, making sure that the alarm goes off at the right time, the ability to snooze an active alarm, and the ability to turn off an active alarm. There's also a few more features that I think would be good to have. Such as a customizable alarm tone, an alarm that gradually increases in volume over time, and the ability to automatically create alarms, a specified amount of time before the first event on a user's calendar, and the ability to have a silent alarm that vibrates only. These P1 features add value and enhance the experience of the alarm. Potentially, auto alarms is something that could be a P0 if we believe that it would be a strong differentiator of our alarm clock app when compared to others, and would provide significant value to users. This is something that we'd want to test and explore it more. These are all still very high level requirements. As I start working with a team, each of these features should be expanded to capture more detail about what it should do. But this is a really good place to start the conversation. Let's talk a little bit about some of the assumptions that I made. For this product, I'm assuming that the user's phone will be powered on. Meaning that if the phone is powered off or the battery's dead, that the alarm won't go off, and that's okay and completely fine. I've also made the assumption that alarms will be based on the local time of the user's phone. The reason that I made this assumption is because I don't want to have to handle time zone changes, so I've listed that as out of scope. How did my PID compare to the one that you wrote?

At this point, you should be able to understand the purpose of the PM role. Understand what a PM does during the different stages of the product development cycle. Identify key cross-functional partners and customize interactions based on understanding their key priorities. Describe different models for gathering requirements. Define the components of a PRD and how to complete each component, including documenting feature requirements. Congrats again on completing lesson 1, and I'm looking forward to seeing you soon in lesson 2.

In this lesson, we're going to talk about how to identify problems. When you're building products, one of the most important things is to make sure that you are solving the right problems. PM spend a lot of time just doing that. They understand the market, the user, business goals, and then identify opportunities. There really is no shortage of problems that could be solved, but one of the hardest things to do is to decide what not to solve. In this lesson, we'll cover the following topics; identifying opportunities, understanding the market, the target user, what total addressable market is, creating and evaluating hypotheses, and building a business case. All right, let's get going.

As you recall from the prior lesson, finding the right problem to solve is critical for the success

of your product. If you aren't focusing on the right problems, your product will fail. This is something that both big and small companies struggle with. You'll have to work through a lot of constraints and trade-offs to get this right. So how do you figure out what problem to solve? Well, there's a lot of different ways that you can get signal. Let's start with research. Research is one of the things that I find the most insightful. There's lots of different types of research that you can do. For example, market research is helpful in understanding the current space. What products or related products already exist? What's the competition like? What are the trends? User research is also super-critical here. There's lots of different forms of user research that can take place, such as ethnographies, interviews, and focus groups just to name a few. But at the end of the day, it's super important to go out and talk to users. This really is the best way to understand their needs and the problems that they're running into. In the case where you might already have an existing product and you're looking for a follow-on product, you already have some additional data that you can look at. You can look at how your current product is being used or not being used. This will give you an understanding of problems that users might be running into with your current product and potentially ideas for new features. You also can look at support data. Again, this will give you a sense for the type of issues that your users are running into. Fixing these issues will reduce bad experiences for customers and it will also reduce the amount of money that the company spends on providing support. Lastly, you can think about efficiency gains. These are things that will allow you to scale and provide more value over time. For example, if there's a manual process your engineering team has to do, if you automate that process, it frees up more bandwidth for the team to solve new problems. There are a few other challenges as well. At the end of the day, time is limited. There's only 24 hours in a day and you have a limited number of people on your team. It's just not physically possible to solve every single problem. Another thing to keep in mind is that not all problems have the same impact. Solving some problems will have tremendous impact whereas other problems might have almost negligible impact. Given limited time in differing levels of impact, not every problem is worth solving. You want to make sure to maximize the impact of you and your team's time. Speaking of impact, it's not just about solving problems for the user. It's also important to keep business goals in mind. If products aren't helping to meet business goals, they ultimately won't survive long term. This could mean that your product gets killed, or even worse that your company goes out of business. Business goals can take a lot of different forums, like increasing revenue, increasing number of users, increasing user engagement, decreasing costs, and decreasing churn, just to name a few. These will likely come up when you build a business case. As you start understanding business goals and evaluating problems, you want to look for overlap between these two. This is where you'll find viable product opportunities. There's a quote from Sam Altman, of Y Combinator that sums this up nicely. "You can change everything in your startup except the market. So spend a lot of time upfront to make sure that you've thought through your market." You can have the smartest most talented people on your team and you could be building amazing pieces of technology, but if your product doesn't solve a problem that resonates with the market, none of that will matter.

Understanding the market is critical to building successful products. Some markets are better than others and some products do a better job than others. You'll want to find product/market fit. So what exactly is product/market fit? Well, product/market fit means being in a good market with a product that can satisfy that market. The next thing that you might be asking yourself is what makes a good market? Let's talk through a few things that go into this. The first is size. The more people or the more need, the more adoption and the easier it will be to monetize your product. The next factor is growth. Is the market growing? Think about how many people have smartphones today compared to 20 years ago. Lastly is acquisition. How easy is it to get new customers? Think about expanding your reach and your ability to explain your product to new users. This can also include getting people to switch from competing products. Think about how much friction there is between switching from an Android phone to an iOS phone, or from changing your e-mail provider from one to another. It can be a little bit tricky to identify product/market fit, but here are a few signs that you don't have good product/market fit. First of all, if users aren't getting value from your product, this will cause a whole bunch of other problems. For example, as a result, they won't recommend it or talk about your product with other people. This also means that there's no excitement about your product. The press doesn't really care, or maybe the articles that are written about your product are boring and bland. Lastly, all of this will lead to very slow growth and make it difficult for you to sell your product because you are not providing any value for the user. On the other side, here are a few signs when you can tell that you've found good product/market fit. It all starts with users getting a ton of value from using your product. It's solving a super important need for them. The product as a result, will practically sell itself. Users will recommend it to others, and it's easy to immediately understand the value and why someone would want it. The press is also super excited, and they're banging on your door to talk to you about your product and learn more. Lastly, because your product is in such high demand, it can sometimes be difficult to keep up. For example, maybe you are constantly needing to add more servers in order to support your app. So how can you get more insight into markets? Well, there's lots of ways to do that. It's important to familiarize yourself as much as possible with what's going on around your product space. This can be done by doing online research to better understand what's happening in the market by staying up-to-date on news that might be relevant to the problem that you're focused on. You can also look at similar products in the space and see how they've approached the problem, and what's working well and maybe where they're struggling. Trends can also provide very important insights. One of the most helpful is Mary Meeker's Annual Internet Trends Report. I read this every single year and there's tons and tons of interesting information. I'll include a link to this report in the classroom. Google Trends can also be a helpful tool to get a sense of what people are searching for and how that is changing over time. So now that you have an understanding of markets and how to get insights about them, let's try it out.

For this exercise, imagine that the team is interested in building a product to help people

better track their fitness and improve their health over time. Spend a little bit of time doing some research to better understand current trends and the fitness tracking market. Here's some questions to get you started. What benefits does improving fitness provide? What products are already in the market? How much money and time do people spend on health and fitness? What are some of the different ways to deliver fitness tracking, and what are the pros and cons of those approaches? Now spend a little bit of time answering these questions and anything else that comes to mind, and then check out my solution in the next video.

So I did a little bit of research for industry trends for fitness tracking, and here's what I came up with. According to Mayo Clinic, there are a handful of benefits of regular physical activity. Exercise can help you to control your weight, whether you have a goal related to lose, maintain, or gain weight. Regular exercise can also help to prevent or manage many different types of conditions, including stroke, high blood pressure, and depression, and anxiety. Physical activity can cause endorphins to be released, which can leave you feeling happier and more relaxed. Physical exercise also helps to boost your self confidence. Regular physical activity can build your endurance and it helps your cardiovascular system work more efficiently, which in turn helps you to have more energy. Lastly, regular exercise can help you sleep better. This includes both falling asleep faster as well as having a higher quality of sleep. So what fitness and health tracking products already exist? Well, there's a handful of smartwatches out there that also include fitness features like the Apple watch and many Android watches. They also include a lot of other features and functionalities as well. Then there's also a handful of smart watches that are much more fitness focus, like the Fitbit and the Galaxy Fit. There's also a handful of fitness related apps. Similar to watches, there are more general fitness apps that do a lot of things like Apple Activity and Health apps and Google Fit, and then there's apps that are more focused on specific activities like Nike Running Club, RunKeeper, and Freeletics. One other product that I came across while I was doing research, that I thought was interesting, is a product called the Motiv Ring. It's a ring that you wear on your finger and it tracks your heart rate activity and sleep patterns 24 hours a day, seven days a week, and it starts at $200. So how much do people spend on health and fitness? Well, a company called Myprotein, surveyed 1,350 adults in the US to age 18 to 65, and found that the average American spends $155 per month or $1,860 per year. When you break this down, they found that people on average spend $56 a month on health supplements, $35 a month on clothing and accessories, $33 a month on gym memberships, $17 a month for healthy meal plans, and $14 a month for trainers. So now let's look at some of the different ways that we could potentially deliver health tracking. One option would be to build specific hardware, like a watch that users would wear in order to track their health. The other option would be to create an app that users can download on their phone to track and record their health and fitness. But there's also an in-between where we could create an app that could run on an existing watch in order to track fitness. So let's look at the pros and cons of these three different methods. There were three different things that I wanted to look at when thinking through pros and cons. The first was investment. How much effort will it require

to build this product? The second is ease of use for the user. What will be the best experience for the user with the least amount of effort on their part? Lastly, market. How big is the market if we were to enter it? For investment, building any type of hardware is a huge investment and requires a lot of expertise across a number of skill sets. Building an app for a watch or a phone now isn't quite as complex. In terms of ease of use, how much effort does the user have to put into use the product in order to get the most out of it? With hardware, it's fairly effortless. As long as the user is wearing it, there's nothing extra that they have to do to get the product to work. With the watch app, there's some minor friction, but if the users wearing the watch, all they really need to do is open the app. With the phone app, there's more friction because the user has to have their phone with them while exercising and they have to open it up. Lastly, in terms of market, I think that building a watch would have the smallest market since it's already a pretty crowded space, and smartwatches tend to be more expensive. In terms of building an app for a watch, it's a little bit bigger of a market, but not everyone has a smartwatch and the app might not work with all watches that exist. Additionally, not all watches will have the same set of sensors, so we might have to get a little bit creative. Lastly, building an app will have what I think will be the largest market because it's something that most people already have and they'd be able to download and use our app immediately. How do these industry trends compare with the ones that you found?

Identifying a target user is important, because it makes it very clear who you're building the product for. Oftentimes, a problem will best be solved in different ways for different types of users. Identifying a target user creates focus, and lets you zoom in on solving the needs of that specific target user. Remember, if you were building and designing a calculator for a student in grade school, it would look very different than if you were building a calculator for a scientist. These users have very different needs, and very different goals that your product would need to help them accomplish. Additionally, they also have different purchasing power. You want to keep your target user in mind, when you're building a product, because it will help you to answer a lot of questions about what your product is, and what it should do. Your target user is who you are building your product for, and in order to define your target user, you will want to identify users with shared characteristics who are likely interested in your product. To do this, you'll need to look at a variety of data like market research. You also can perform interviews, or conduct surveys and run focus groups. If you already have customers, you can look at existing customer data, and there's lots more. Once you have this data, you'll want to look at user characteristics for trends in the following types of things. Demographics, for example, how old is your user and where do they live? Motivations, what gets them excited? Goals, what are they trying to accomplish? Frustrations, where are they running into current problems and challenges? Once you've identified trends across these characteristics for your users, you can create user personas to represent them. A user persona is a fictional person that you create who might use your product. As you're developing the product, you'll refer back to your personas, in order to stress test and challenge some of your ideas. For example, does this decision solve Sam's frustration, and would Sam be able to afford this

product? Let's walk through some of the elements that make up a persona. You'll want to start by creating a name for your persona. Personally, I like to use alliteration, and have the last name describing an attribute of the user like Sam Student. Another example would be Dan Driver. It's a nice touch to include a photo, or illustration of what this user might look like, as well as a quote with a key insight about this user. In this case, the quote I'm using is, "I miss going on my daily morning run." Keep in mind that it can be really really powerful to incorporate real quotes from users that you've talked to during user research. You'll also want to include a description of that user. This will paint a little bit more of a story about who they are. It doesn't have to be super in depth, but you'll want to keep the details focused on things that are relevant to your product. For example, Sam is just starting her first job out of school, and lives with two other roommates. She loves being outside and is a big runner, but after adjusting to having a new schedule, she has found it challenging to make time to exercise on a regular basis. You'll also include frustrations, or challenges that the user is facing. In this case, Sam's new schedule makes prioritizing exercise difficult, and she also feels intimidated to get back into going into the gym. Then, you'll also want to outline the goals that the user has, which your product should also help them achieve. In this example, Sam's goals are to exercise more consistently, and to be able to complete an upcoming marathon. Lastly, you'll want to include a few things that motivate the user. These can be helpful things to consider when designing your product. In Sam's case, she's motivated by being able to see her progress and improvement over time, and she's also motivated by being able to share progress with friends and family, in order to get support from them. That's a quick intro to target users, and how to create user personas. Now, you should give it a try.

Given the market research that you've done so far, create three user personas for a fitness tracker. Keep in mind that in the real world, you would also want to make sure that you get data from real users through interviews or surveys before building any personas. For each persona, create a name and some information about that person. Also include their motivations, goals, and frustrations. Then we'll regroup and I'll walk you through the examples that I created.

Now, I'm going to walk you through three personas for potential users that we could target. John is our first target user. He is 37 years old and has been increasingly focused on work over the last few years. As a result, he hasn't been as active and he's put on some extra weight. For the new year, he made a resolution to lose 20 pounds during the year. John's frustrations are that work is stressful and keeps him busy. Another challenge that he's running into is the activities that he finds relaxing don't necessarily help with his health goals. For example, video games and eating out. John's goals are to lose 20 pounds during the year to more consistently exercise and to find activities that promote good health. So I'm going to go ahead and summarize John's persona with a quote that, "My new year's resolution is to lose 20 pounds." Sally is our second target user. She is 27 and really enjoys running. It's one of the few things that really helps her to disconnect. She's run a few shorter races and

recently completed a half marathon. She is really interested in a full marathon but wants to train to make sure that she can complete it without injury. Sally's frustrations are that she wants to make sure that she can finish a marathon successfully. She is also a little bit worried about getting injured during the race. So her goals are really to just complete a marathon and to avoid getting injured while running. I'm going to summarize Sally's persona with a quote, "I want to run my first marathon." Chris is our third and final target user. Chris is 32 and has been running competitively for the last 10 years. Recently Chris feels like things is stalled out and he is not seeing improvements. Chris' time has remained about the same. Chris' major frustration is not seeing an improvement in terms of speed, and the goal that Chris has is to beat his personal best record. So I'm going to summarize this persona with that quote, "I want to be my best race time." So those are the three personas that I created for our target users. How do they compare to the ones that you created? Specifically look at the frustrations that I identified and the goals as well. Were they the same or different and the ones that you identified for your target users?

Now, we're going to talk a little bit about TAM or Total Addressable Market. TAM is a measure of the revenue opportunity for our product. Keep in mind that TAM is not a measure of your revenue or future revenue, but it allows you to understand the size of the market if you had 100 percent of the market. A larger TAM indicates a larger opportunity with more demand for a particular product. However, just because there's a large TAM doesn't mean that your product is guaranteed to be successful, there's lots of other factors that will come into play, like competition. TAM is defined as average revenue per user times the total number of potential users in the market. There's a couple of different ways to measure TAM, and we're going to walk through three of the most common. First, there's top down, then there's bottom up, and finally, there's value theory. Let's get started with the top down method. The top down method uses industry research to estimate the size of an addressable market, this approach starts from a high macro-level view of the economy. You start with the total number of people in the world, and then you narrow down that number based on factors like demographics. This goes all the way down to your target user. These calculations can be quite complex, but you can also build a simple top down TAM using stats that are available. Let's walk through an example to find the TAM for bottled water in the United States. We know that there's roughly 7.8 billion people on Earth, and we know that there are roughly 330 million people that live in the United States. We also know that each person needs about two liters of water per day, and we know that we're going to charge one $1 liter. There's 330 million people in the United States and they each need two liters of water per day. We're going to charge $1 per liter, and there's 365 days in a year. This works out to a total TAM of $241 billion. The second method for measuring TAM is a bottoms up approach, this involves using known data points that you have, this could come from early customers and sales, although you could potentially use theoretical customers and sales as well, and then extrapolating them to represent a larger market opportunity. Imagine that we're already selling bottled water in the State of California, and we're interested in expanding it to the rest of the United States. But

before we do this, we want to understand the TAM. With the bottoms up approach, we can look at our existing sales and scale it. In this example, we're still selling water for $1 a liter, the price is exactly the same. But we know that we sell 10 million liters every day, we also know that the population of California is about 40 million people. This means that we sell one liter of water per day for every four people. We know that there are about 330 million people in the United States. So switching out the population, we get that the TAM for the United States for bottled water is $30.1 billion, this is a pretty big difference compared to the top down approach. In the top down approach, we used some pretty crude assumptions, whereas in the bottoms up approach, we already had some insight into how many people would actually consider purchasing bottled water. The last approach is the value theory approach and it's best used for brand new product categories where you don't really have much information to base any of your estimates on. There's two main questions that you need to answer. How much are people willing to pay for the product? How many will purchase the product? To get these numbers, you'll have to do a lot of research and talk with users to see what they would be willing to pay and to figure out an estimate for how many potential customers you have. We just talked about what TAM is and how there's three different methods that you can use for calculating that. Now, you should give it a try.

Now that you've learned a little bit more about what TAM is and how to calculate it using different methods, it's your turn to try. I'd like you to calculate the TAM for streaming media services in the United States. Then after you've done the calculation, check back to see how your answer compared to mine.

So now let's walk through the calculation that I did. There's 330 million people in the United States, and I'm assuming that streaming services is something that most people will be interested in or you use at some point. Similar to how traditional television has penetrated the market. Most streaming services costs around $10 or so a month. So that's the number I'm going to be using. So if we take 330 million people and we multiply it by $10 per month and 12 months in a year, we end up with a TAM of $39.6 billion. So let's refine this a little bit though. Streaming services are generally sold and shared within a household, not to individual people, similar to how a cable TV subscription would work. So there's roughly a 130 million households in the US. The $10 per month price point will stay the same. But if we update the equation to use a 130 million households at $10 per month and 12 months in a year, we end up with a TAM of $15.6 billion. How did these numbers compared to the ones that you calculated? Were they higher or lower? Or maybe even the same.

Understanding the ROI or Return on Investment of solving specific problems is critical. As I've mentioned before, there's an infinite number of problems that you and the team can be solving. But there's only so many hours in the day. Understanding and calculating ROI for different problems will help you to understand where to focus your team's time in order to have the biggest impact. ROI is a way to measure the efficiency of an investment, like

deciding to build a new product or feature. In a mathematical sense, it's the amount gained minus the amount spent divided by the amount spent. Amount gained doesn't just have to be an increase in sales, but it could also be cost savings. You just want to make sure that you get the impact into something that can be measured in dollars. Keep in mind that these are just estimates, and they can get pretty complicated when you start adding in lots of factors. You should work with someone on the finance team or the data science team to help build this out in more detail. One thing that is important is that the higher the ROI is, the better the business outcome. This is because high ROI means that you're able to create more impact with less effort. For many software products, it'll look more like this; Impact minus the cost of development time divided by the cost of development time. We can estimate the cost of development time based on how many people are involved and how long it takes to complete the project. Imagine that there are four problems that you want to solve, each problem has a cost and you have $5 that you can spend on solving problems. You've also estimated the impact of solving each problem. What are the right problems to solve? At first glance, problem A looks like the best option. It will cost you all $5, but you'll get $18 in return. But, make sure you look at solving problem B and D. Doing this would only cost you $4 and you would get $20 in return. You'll still have $1 left over when you compare it to spending the $5 on solving problem A. Now, let's go ahead and calculate the ROI for problem D. So remember, this is the formula for calculating ROI. So let's find the ROI for problem D. The impact is $10 and the cost of development is $1. So $10 minus $1 is $9. Then dividing that $9 by the cost of development, which is $1, is 900 percent. So after calculating the ROI for all of these problems, we can see that solving problem D has the highest ROI. Remember, we have $5 to spend on developing products. Problem D will take $1 to solve. After solving problem D, we have $4 left. Problem A has the next highest ROI, but we only have $4 left. So the next project that we should fund is solving problem B. There's one other thing that can be helpful when understanding ROI, it's called the payback period. The payback period is the amount of time that it takes for you to get the cost back from the product. You can calculate the payback period by taking the cost, and then dividing it by the impact of the project, divided by the amount of time that it takes to realize that impact. Let's go ahead and add how long it will take for the full impact to be realized. Then the first thing we're going to do is calculate the payback by a year. For problem A, the impact is $18 over three years. This works out to a payback of $6 per year. Here you can see that we've calculated the payback per year for each project, and because we know the cost of the project, we can also calculate the payback period by dividing the cost by the payback per year. In this case, $5 divided by $6 per year ends up being 0.83 years or 10 months. The payback period is also an important factor to consider when deciding what problems to solve. Calculating ROI and payback periods are two important tools to help you understand if it makes sense to solve a problem. Projects with positive ROIs will cover the investment required to build the product. Projects with negative ROIs will not and won't make sense to pursue. Calculating the payback period allows you to understand how long it'll take before the product recoups the upfront investment that was required to build it. It's important to understand the ROI and payback period for your

product, especially as you get closer to building a business case for it.

Before you build a product, you want to make sure that it will be successful. To make sure that your assumptions about your product are correct. For example, that you're solving a real problem for real people. You can create hypotheses based on what you know so far and then test them to see if they're still accurate and hold true in the real world. There's a couple of different types of hypotheses that you'll want to test focused on the following aspects, have you identified a customer need? You'll need to take into account what are the specific goals of your customers that they're trying to reach and where is there an opportunity for you to be able to help them reach those goals and meet their needs. Does your solution meet that need? This will depend based on exactly what your product does or doesn't do. Is your solution differentiated? How does your solution compared to other solutions out there? Does your product create enough value for users? When thinking through hypotheses related to customer needs, the question you're really trying to answer is, is this a real problem? Is there urgency in finding a solution for this problem? Think of a painkiller versus a vitamin, painkillers provide immediate relief. Vitamins for the most part, are just nice to have. Does this problem require a painkiller or are you just building a vitamin? If there's no need or urgency, it's likely that the product will fail. When creating hypotheses related to your solution, the question you're trying to answer is, does your solution really solve the problem? It's also helpful here to think through what are the alternatives that the user has if they don't go with your solution? How does your solution compare to everything else out there? Is it about the same or is it way better? For differentiation, the question you're trying to answer is, is our product the same or different? How is it different? What does it do that other products don't? Lastly, for value, the question you need to answer is, does your solution create enough value for users that they're willing to pay for it? If so, how much? It can also be interesting to think through what would increase a user's willingness to pay, especially if the answer to the last question was no. It's also helpful to think through how much do people currently pay for alternative solutions. So we just talked a little bit about the different types of hypotheses that you can create and how they can be important and helpful. In the next video, we'll talk about how you can test them.

Now we're going to talk a little bit about testing a hypothesis. You can test and evaluate your hypotheses in several different ways. You can do this through user interviews, where you go and talk to your users, walk them through your thinking and get their thoughts and feedback. This works best when you have a prototype that you can show to users and ask them to evaluate. It's also helpful to get a sense if this is something that they would use and if it's something that they would consider buying, why or why not. This will help you to understand if you're solving a problem for users, as well as how users are perceiving your solution. You also can get feedback from focus groups. You'll get a group of users together in a room at the same time, and then you'll have a conversation with all of them. It's a little bit different than one-on-one interviews because participants in a focus group will interact with each other, and

they'll build on top of each other's ideas or disagree and provide alternative views. One thing to watch out for here though is group thinking. This is where there's only one opinion that surfaces within the group. You can also run surveys for some of the higher-level questions you might have. Surveys are great when you want to get feedback from lots of people, but you are a little bit more limited in the types of questions that you can ask and how deep you can go. Think of having multiple choice versus being able to have an in-depth conversation about a topic. You'll get some insight, but it won't be as detailed as what you would get from an interview. For all three of these options, it'll be super-helpful to work with a researcher who can help you to set this up and structure it in a way that will minimize bias. You can also put your hypotheses into a design sprint. This will help to stress test the way the problem is framed and further define a solution. The last phase of the design sprint is validate, where you'll put concepts in front of real users and get their feedback. As you're going through this process, if you find that you're not addressing a customer need, or your solution isn't providing value, or your product isn't differentiated enough, or there's no willingness to pay, stop. It's not worth pursuing this. You don't have product-market fit. You should go back and tweak which customer you're focusing on, or your solution, or your pricing model. But at the end of the day, you'll want to make sure that you believe in your hypotheses and that you have the data to back them up before you start building a product.

Once you've validated your hypotheses and have confidence that there's something there, it will be important to build a business case to justify why the team should spend the time solving the problem that you identified. A business case answers the question, why should the company do this and invest in it? The goal is to convince sponsors and stakeholders to approve the project. You don't always need to go to this level of detail for everything though. For example, adding a new feature to an existing product. In this case, for smaller projects, the PRT will likely have enough detail, but for an entirely brand new product that you're going to build from the ground up, you will need to be able to justify the larger investment. Is it worth continuing this project? Doesn't make sense? In order to help executives make this decision, these are the types of things you'll want to make sure are in the business case. You'll want to clearly outline the business problem, what the benefits are to solving the problem, the associated costs and risks, as well as possible solutions, and a timeline. It's also super helpful to provide an overview of what the competition is doing. Now let's walk through how you can structure all of this in your business case document. These are the components, executive summary, problem statement, analysis of product opportunity, possible solutions, and recommended approach. Let's go into more detail. For the executive summary, it should summarize the entire document, including the recommendation. If people only read this and walkaway, they should understand exactly what it is you're trying to do. It's also the first thing that people read, but usually it will be the last thing that you write to make sure that everything is cohesive. For the problem statement, you want to give an overview of the problem, and why it matters. Why is this something that's important to solve? What is the goal behind solving this problem? How does solving this problem link back to business

strategy and business goals? In the analysis of the product opportunity, you want to answer the question, why is this important for the business to address, and what is the cost of the problem or the size of the opportunity? For possible solutions, you'll want to detail each possible solution. For each one, you should discuss the benefits, the costs in terms of people, time or money that will need to be spent, as well as a timeline and risks. When you're putting together this information, it's super important to use as much data as possible. This will help to build confidence in the business case. For recommended approach, here you want to explain which approach you are recommending as the best one and why. Now I am going to share a few tips about putting together a business case. You'll want to tell it to your audience. Knowing your audience is super important. You should be aware of who you need to convince and get approval from and tailor the business case to be super relevant to that person and make sure that you're answering all of their concerns. It's also important to use language that's familiar with your audience. Keep it short and concise. There's definitely a lot of stuff that you need to cover, but the key is to only include relevant information that will be helpful to making a decision. You can always include links to other documents if people want more information. It's also going to be important to make it interesting. You want people to walk away feeling super excited about solving this problem. Lastly, make sure you demonstrate the business value. Again, the more data that you can provide here, the better outcome that you're going to have because it creates confidence. We just talked through what a business case is, what are the components that go into a business case, and some tips that you can keep in mind when you're putting one together.

At this point, you should be able to: identify problems that are worthwhile to solve while thinking through market research and user research; understand the market through deeper research and identify good markets based on size, growth, and acquisition, as well as identifying product-market fit; identify your target user which has shared characteristics and are likely interested in your product, as well as being able to build user personas that include frustrations, goals, and motivations; calculate the TAM for your product, as well as understand three different methods for calculating, including top-down, bottoms up and value theory; calculate the ROI for solving a problem and the associated payback period; define hypotheses about your product that need to be validated around customer need, your solution, differentiation, and value; test your hypotheses by putting them in front of users, either through user interviews, focus groups, surveys, or by running a design sprint; and lastly, understand the components required to build a business case, so that it's easy for stakeholders to understand the size, impact, and importance of the problem. Congrats on completing Lesson 2. I'll see you in the next lesson when we talk about vision and strategy.

In this lesson, we're going to talk about vision and strategy. Specifically, we'll cover the following topics: Vision; this is about ultimately defining what it is you're trying to build. Strategy; this is about how you're going to get there. Business models; mapping out all of the components that will be required to build your product for your customers, as well as different

revenue models for capturing value and competitive analysis in terms of building and understanding of other products that are out in the market and how your product compares to them. MVP or Minimum Viable Product; this is about understanding what's the simplest product that you can get to market the fastest, and KPIs, or Key Performance Indicators; how do you measure your progress and success. Let's go ahead and get started.

Let's talk a little bit about vision and just what exactly it entails. Vision describes what your product will look like when it gets to its final state. It explains the essence of your product, what it'll do, and why it matters to your users. Generally, your vision should be able to be summarized in a sentence or two with an ambitious goal. For example, the vision for Google WiFi was to create a wireless access point that people love. This is a simple and powerful vision and it was something that guided every single decision as we were building the product. But vision is more than just coming up with a sentence or two. You need to do the work to create a compelling story that answers what you're building, who it's for, and why it matters. You'll need to leverage a lot of the research and insights that you've already done in order to do this. But remember, it's best to keep your vision high level. You don't need to get super into the details of exactly what it'll look like. Instead, just focus on what it will do at a high level. Vision is really important for a couple of reasons. First, it gets the team excited and motivated about building the product. Secondly, vision serves as a north star to guide the team and decision-making as your product is being developed and as the north star, everything that you do should trickled down from your vision. This includes strategy, which is about how you're going to get to your vision. Will talk more about strategy in a little bit, as well as the features that are on your road map. Because vision will impact every single aspect of your product. It's really important to be thoughtful when you're crafting your vision.

So let's talk about some things that you should keep in mind when you're crafting your vision. You want your vision to be inspiring. You want your vision to get the team excited about the product. I've found that a really good way to do this is by focusing on the problem that you want to solve for users, and why it matters. How will your product change their day, their month, or even their life? Steve Jobs once said, "If you're working on something exciting that you really care about, you don't have to be pushed. The vision pulls you." You also want your vision to be ambitious. Think big. You want your vision to be broad and open-ended, to challenge the team to think about the product in new ways. It shouldn't detail out exact flaws or features. You'll work with the team to come up with those later. Additionally, by being ambitious, it gives you a little bit of leeway so that you can change, adapt, and pivot the product as you learn new information while still being able to work towards your overall vision. You also want to make sure that your vision is easy to explain, and people should be able to get it, and understand it when they hear it. Generally, this means you'll want to be able to summarize your vision into a short, concise sentence, or two. All the other details are important, but they don't need to be part of your high level vision. Those more detailed things will show up in other places and documents that you put together. It's also important that your

vision is something that the team believes in. You'll want to make sure that the team believes in the vision that you're creating. You can get to this point through a lot of different ways. But I've found that involving the team early on works really well. Potentially this could look like asking team members to participate in some of the early research that you'll be doing and talking with customers. Alternatively, you could also go through a visioning exercise together as a team to co-create a vision. Your vision needs to evolve over time. It's really hard to get things exactly right the first time, and it's okay if you don't. But what's important is that as you learn new things, you take that information into account and evolve. Lots of times it might be necessary to make a pivot into a new direction. The important thing is that you realize when this is the case. Your vision should also be something that you share out frequently so that everyone on the team understands your vision and is on the same page. The team will end up using the vision that you create to set up an inform decision-making. You'll also want to tailor the way that you present the product vision to different groups based on the things that they care about the most.

So now we're going to talk a little bit about strategy. What exactly is strategy? Well, it's really about how you get to your vision. Strategy is all about how you're going to turn your vision into reality. When you're building strategy, there's many, many factors that you need to take into account, such as user needs. It's really important that you understand who your target users are and what are their needs that could be solved by your product? Key features is another important area. Here you'll want to spend time thinking about what's the right mix of features and functionality that will meet your user's needs and create value for them? Additionally, are these features things that will result in people using your product, and will these features result in people being willing to buy or pay for your product? Understanding competitors and how your product can be differentiated is also important. This means understanding on a deeper level what are competitors doing and how is your product different from those competitors? It's also important to understand how your product can help to achieve business goals. In order to do that, you have to understand what are the needs and goals of the company. Usually, this will be related to revenue, but it's not necessarily always the case. The other thing that you need to think about here is how can you measure these goals? Lastly, it's really important to be aware of and understand trends. For example, how is the market changing over time, and how are user behaviors and needs changing over time?

So now that you know what strategy is, let's walk through a few things that you should keep in mind when you're creating your strategy. It is important to find the right balance between features, user and market needs, and business goals. You need to find the right balance between what features to build, the needs of your users in the market, and the business goals you need to achieve. Some features that you could build won't provide a lot of values to users and some features won't help meet business goals or might be too expensive to build. You should also think of your strategy as a way to create building blocks for how to get to your vision. Think about breaking your strategy out into 3-5 smaller components that you need in

order to realize your vision. These are the things that your product needs to get absolutely right in order to be successful given the vision, what you know about your users, the market, and your business goals. Strategy should also still be high level, but with a little bit more detail. As you're building out your strategy, you want to stay high level, but you should start filling things in with a little bit more detail than what you would get from your vision. The full details though will emerge in [inaudible]. It's also helpful when your strategy is goal-oriented and preferably measurable. Goals make things very actionable because you know exactly what you need to do. It's even better when you can measure them and your progress towards meeting them.

In order to build a viable product, you need to have a business model. What is a business model exactly? Well, a business model describes how a business creates, delivers and captures value. We'll break this down further using the Business Model Canvas. The business model canvas is a helpful tool to map out a business model which was created by Alexander Osterwalder, and it captures a number of different components that will impact your ability to create and bring a product to market. On the left, we have things that need to happen in order to build the product and the associated costs with building and delivering the product to the customer. In the middle, is the value prop. This is about what the product is and why people would want it, and then on the right we have customers. Who your customers are, how you interact with them and how you get your product to them, as well as the associated revenue that your customers generate in exchange for your product. Let's walk through the business model canvas in more depth in the next video.

Now we're going to walk through each component of the business model canvas. As we're going through each component, we will also map it out with an example for Zoom, which is a company that creates software for video conferencing. First up is key partners, key partners help build or deliver the product to users. These could be re-sellers or other partners like payment processors. Looking at Zoom, their product is mostly software based. But they do work with partners to outfit conference rooms with cameras, TVs, and tablets, and they also work with system integrators to integrate their technology into existing spaces. They also work with payment processors in order to collect money from their customers. Next is key activities, key activities are the things that need to happen in order to build your product. For Zoom, there's two big activities, development of that technology and sales and support. Next, we have key resources. Key resources are things that you need in order to deliver value to the user. This could be people, Intellectual Property, financial backing, or any other type of resource. Going back to Zoom, data centers are important in order to deliver the streaming video to participants. Zoom also likely has a compression algorithm that allows them to deliver high-quality video while reducing the amount of bandwidth required. Next, let's talk about the costs that are associated with building the product. This could include both fixed costs, like salary of the development team, and variable costs that increase as the product grows, like data center capacity. It's also important to think through which activities and

resources are the most expensive. So going back to the Zoom example, I think that the two biggest costs will be people and data centers. Next is value proposition. The value prop explains why someone would want your product. In Zoom's case, their value prop is that they built a video conferencing product that works really, really well. Their product also works on phones, laptops, and in conference rooms that have TVs, and it can scale. It can support over a 1,000 participants in a call, which is crazy to think about. The next component is customer relationships, think of this as how you are going to build a relationship with your customer, including through interactions in the product. You'll want to make sure to think about building relationships with both the person using the product and the person who purchases the product. This is often the same for consumer products, but can vary for Enterprise products with Zoom at the end of every call, they asked the user to rate the call. This allows the team to get insight about how the product is performing from the customer's perspective, and as a customer, it also makes me feel like if I had a bad experience, that someone is going to be looking at that data to improve their product in the future. For Zoom, it will also be important to think about how to build a relationship with the decision-maker who decided to purchase the product. In this case, it's likely the IT department. Next is channels. Channels is all about how you get the product to the customer. It's also important to think about which channels work best and which are the most cost effective. Going back to the zoom example, there are three channels that I noticed. They had their website where people can sign up and purchase the product directly. They have referrals as well, so for example, if I'm a Zoom customer and I invite you to a Zoom meeting, you learn about the product through my referral, and lastly, they have resellers who sell their product on their behalf. The next component is customer segments. This is all about understanding who are your customers and which customer is the most important. In the Zoom example, their customer is really distributed people or teams who need to collaborate and communicate. This is really, really broad and you can split it into further segments such as SMBs, enterprise, education, healthcare and even government. Lastly is revenue streams, what are customers willing to pay for how much and how do they pay. For Zoom, the main revenue stream is through subscriptions. There's several different tiers of subscriptions with more expensive subscriptions offering more features and functionality. We just walked through all of the components of the business model canvas as well as created an example canvas resume. I know it was a lot to cover and walk through, but the business model canvas is a really helpful tool to map out and understand all of the components that you need to build a product and a successful business.

Now we're going to talk a little bit about different types of revenue models or ways that you can sell or monetize your product. There are lot of different revenue models out there, but we're going to walk through some of the most common, starting with ads, which in that case, advertisers pay money in order to show relevant ads to your users. There's also purchase or licensing for software, where your users pay a onetime fee in order to purchase the product. There's pay-per-use, where your users pay every single time that they use the product.

There's subscriptions, where your users pay on a continuous basis for access to your product. Lastly, there's a freemium model, where your product offers a limited number of features for free. Note that sometimes these can overlap. For example, you could have a freemium model supported by ads with the goal of users converting to a subscription. So in the freemium model, think of this circle as your product. You will give access to some of your product at no cost to the user. This is potentially something that could be supported through ads. But the ultimate goal is that users will convert and purchase your product. So let's dig in a little bit deeper to each of the revenue models that we just talked about. In products that are ads based, you generate revenue by showing ads from third parties. The more effective your targeting is to show relevant ads to interested users, the better this model will work. Advertisers generally pay for ads on a CPC or cost-per-click basis. This is much more common in search products or they can also pay for ads on a CPM or cost per 1,000 impressions basis. This is more common outside of search products. For ads, revenue comes from ads being displayed or clicked. This means that the more users you have using your product, the more clicks and ad impressions you'll generate. Click-through rate is an important metric of how often users are clicking through on ads in your product. Some examples of products that are supported by ads are Google Search, Facebook, Amazon product ads, which are listings that show up in search results when you're searching for something to buy on Amazon and Spotify free. Freemium products have a free offering that's available at no cost to the user. There's generally some limitations or restrictions involved with these types of products. But if users purchased the paid tier, those restrictions go away. The freemium model is an easy way to get users to try the product and let the product sell itself. Some examples here include Spotify, Zoom, LinkedIn and DropBox. There's also software licensing. When a user purchases a license for your software, they pay a onetime fee and in return, they get the right to use your software. Your revenue is based on the total number of units or licenses that are sold. Some examples of these types of products include traditionally Microsoft Office or Adobe Creative Suite, although both of these products are now offered on a subscription basis as well. Another model is pay-per-use. Each time that the user uses the product, they pay for the amount of the product or service that they use. Revenue is generated when people use your product. So the more your product is being used, the more revenue that you'll generate and earn. Examples include Uber , DoorDash and Fandango. Lastly, there's subscriptions, where users pay a recurring fee in order to access the product. It's usually monthly, but not always. This means that revenue comes from paying subscribers. This in turn creates a recurring revenue stream, meaning that as long as you maintain your subscriber base, you will generate revenue on a regular basis. Some examples of subscription-based products include Netflix, Spotify Premium, Microsoft Office 365 and Zoom. So we just walked through some of the most common types of revenue models. But keep in mind that this wasn't an exhaustive list and there's lots of others out there.

We just walked through all of the elements that go into making up a business model canvas,

as well as a few different types of revenue models. Now, I want you to take what we learned and start thinking about how to map out a business model canvas for the fitness tracker.

Let's walk through the solution, that I put together for the fitness tracker Business Model Canvas. Let's start by talking about our customer segments. This product is really for people who want to improve their fitness. Specifically, we're going to focus on people with goals around weight loss, or training to compete in a competition, like a marathon. Now that we've identified our customer, let's talk a little bit about the value proposition for this product. The core value prop is to help users improve and track their fitness over time. Specifically we can help users improve their fitness by connecting them to a virtual trainer who can help to customize workout plans. We're also going to create a feature that allows users to track and share their progress with friends. Since this is going to be an app, and not a piece of hardware, the key activity to build this product is software development. But for key resources, we'll also need to make sure that we have a fitness specialist on the team, who can help us to understand the science behind different types of exercises in order to achieve certain fitness goals. For key partners, we'll need to include fitness trainers who will be matched to pain users in order to provide tailored coaching. But thinking through all of this, there's also an opportunity to partner with gyms, where you could potentially be matched with a trainer at your local gym. Additionally, there's also the potential to build an integration with some exercise machine manufacturers. This would make it much easier for users to capture their workout data directly from the machine. So looking at the things that we have so far on the left side, we can see that there's a fixed cost for the people who are developing the product, as well as a variable cost based on the number of trainers, and partnerships, and usage. So going back to our customers, we're going to build a relationship with them by sending out a weekly progress summary. This summary will highlight the progress that they've made towards reaching their own specific goals. In terms of getting our product to users, there's three main channels that I've thought of. Users can directly download the app from our website. They also could potentially be invited to the app by a friend, either from receiving shared progress, or by being invited to a challenge to see who could reach a goal first. Lastly, if we end up partnering with gyms, we can also use those gyms as a channel to acquire new users. Lastly, in terms of generating revenue streams, the product will have a free tier that's supported by ads, as well as a paid subscription, which is required to unlock individualized coaching from a trainer. Now that I shared the Business Model Canvas that I created for the fitness tracker, take a few minutes and think about how it compares to the one that you created.

So now we're going to talk a little bit about competitive analysis. Competitive analysis in its simplest form is identifying other similar products in the market, including the users that each product is targeting. Once you've identified all of the competing products, you'll want to try to understand each product's strategy. You can usually get a glimpse of this from using the product and looking at marketing materials. From there, you can identify the strengths and

weaknesses of each product, what each product does really well, and where the product might run into issues. Having this insight allows you to better predict shifts in market behavior and trends. Because competitive analysis gives you a view into all the other key players in the market, you'll understand exactly how your product fits in, which in turn can help to inform your product strategy, and how you can create advantages for your product. So what exactly does competitive analysis include? Well, competitive analysis can look at a lot of different factors, including things like funding, support, and global presence, just to name a few. But for product insights, I've found that it's most helpful to focus on the product offering and key features. What types of products does the competitor have and what are the key features that make this product attractive to customers and stand out? It's also important to think about differentiators. What makes the competitors different from others that are in the space? As well as identifying who competitors are targeting. Who is that product really intended for? How does the product get to the customer? What are the distribution channels that the competitor has? Lastly is price points. How exactly is the product priced, and how does that compare to our pricing strategy? So how do you go about starting to build a competitive analysis? Well, there's lots of different ways that you can get insight into competing products. The first place I usually start is I Google in order to get a better understanding of just what products exist in or out there. After I spend a little bit of time searching, I'll review websites and marketing material to get a better sense of if the product is a competitor or not. If it is a competing product, I'll generally want to try using it to really understand what it is and how it works. But you can also get a good sense of how well it's doing by reading reviews. Both reviews that are written directly by customers, as well as reviews that you might find in a tech blog. Lastly, it's also really eye-opening to check out the associated social media presence. So let's talk a little bit about the different types of competitors. There are three different types of competitors. If you think about products as doing a job for a user, you have a primary or a direct competitor, and in those cases, the products are doing the same job, mostly the same way for the same set of users. So think of Android and iOS. That's a great example. They're both operating systems for mobile phones. You could also think of Netflix and Disney as being primary competitors. They both offer streaming entertainment services, or Uber and Lyft, which are both ride-hailing apps. The next type of competitor is a secondary or indirect competitor. For these type of competitors, the products do the same job but in a different way and generally for a different type of user. So an example of indirect competitors would be Facebook Messenger and Slack. These are both messaging apps that you can use to stay in touch and communicate with other people. But the difference here is that Facebook Messenger is more designed for personal usage, whereas Slack is really designed for business usage and collaboration with teammates in a work environment. Another example of indirect competitors is coffee versus tea. These are both beverages and they both have caffeine, but people tend to have a strong preference for one versus the other. For example, if you really like coffee, tea is probably not a product that you would consider purchasing. The last type of competitor is a tertiary or replacement competitor. In this case, the product does a different job, but it ends up competing for the same users. It's called a replacement

because it has the potential for consumers to replace one product with another one. A really good example here is the DSLR versus the iPhone. The DSLR camera was designed to take high-quality photos, whereas the iPhone is a device that does a lot of different things, but it also has a camera. Over time, the iPhone has actually started to replace many types of digital cameras. In fact, in 2015, the iPhone became the most popular camera on Flickr. So we just walked through different types of competitors and a couple of examples of each one. When you're building your product strategy, understanding the competition is super important. So it's something you'll always want to keep in mind.

So now that you understand what a competitive analysis is, I want you to spend a little bit of time doing some competitive analysis for the fitness tracker. For this exercise, use the linked spreadsheet template. You'll need to identify three to five competitors and then evaluate the competition across the following dimensions: Product offering and key features, differentiators, target customers, distribution channels, and price points. Give it a shot, then check back in the next video to see the analysis that I did.

So now I'm going to walk through the competitive analysis that I put together for three competitors that are in the fitness tracking market. First, let's talk about the Apple Watch. In terms of offering, the Apple Watch is available in a number of different styles, including different case colors, different case materials like aluminum, stainless steel, or ceramic, and GPS or cellular capabilities. Key features include the Activity app which automatically monitors your movement while you're wearing the watch. It also has the ability for you to manually interactivity. The Activity app gamified being active by monitoring across three different dimensions. Moving, exercising, and standing. It presents this in the format of a game to the user, where you have to close to the rings each day. In terms of differentiators, the biggest differentiator here, is that the watch will automatically monitor your activity while you're wearing it. There's nothing special you need to do. For target customers, well, you have to have an iPhone in order to set up and use the Apple watch. Fitness minded individuals, is probably one of the target users for the Apple Watch. Although there are other target users who could also be considering an Apple Watch for other reasons as well. In terms of distribution, the Apple Watch is available through the Apple Store, both online and in retail stores, as well as through most tech retailers like Best Buy and Amazon. The watch starts at $399 for the most recent model and increases in price based on it's specific configuration. Strava is the next product that I want to talk about. The product offering, is a free app with three different add on subscriptions. Key features are really tailored around tracking and monitoring performance, specifically when it comes to running and cycling. One thing that's unique and different about Strava, is that there's an ability to share posts about your activities and goals with others. In fact, Strava markets themselves has the social Network for athletes. In terms of target customers, Strava is really geared towards the serious runner and cyclist. Although it's still a great product for people who are just getting started. In terms of distribution, you can download Strava through the Apple App Store or the Google

Play Store. They also have a version that's available for smartwatches as well. In terms of pricing, well the core app is free and there's three different add-on packs that are available for two dollars each month. These packs unlock additional functionality for training, safety, and deeper analysis of your data and workouts. Lastly, let's talk about MyFitnessPal. This is another freemium app, with a free tier that's supported by ads, and a paid tier without ads and with additional features. In terms of key features, MyFitnessPal is really geared around nutrition and activity tracking with specific goals around weight management. There's also a food diary that allows you to track what you're eating and when. and lots of tools to easily get nutrition information about what you're eating. For example, by looking up an item by scanning its bar code. One of the biggest differentiators that MyFitnessPal has, is that it has an extensive database of 11 million different foods and their calorie and nutrition information. The target user for MyFitnessPal, is someone who's focused on managing their diet and wants to better understand and track what they're eating. Oftentimes, they might have a weight management goal attached as well. Again, in terms of distribution, this is an app and it's available on both Android and iOS through the respective app stores. In terms of pricing, there is a free version available, but the premium subscription is $10 a month, and unlocks additional functionality, including nutritionist approved recipes and meal plans, as well as additional tracking and goal setting on a more granular level like cholesterol and protein intake. So here's an overview of the three competitors side-by-side. Did you identify any of these competitors? If so, is there any overlap with what you identified and what I identified?

Now we're going to talk about MVPs or Minimum Viable Product. An MVP is just enough features to get early adopters excited, and after you launch an MVP you'll get a ton of feedback that will help you understand if you have product market fit and what areas you should invest in next. So one thing that's super important to keep in mind is even though it's called an MVP, it doesn't mean that the product should feel incomplete, unfinished or unpolished. It's still very important to deliver a high-quality experience to your users, and investing good design and reliability. Remember, think of an MVP as a starting point for your product to grow into something more. If you don't start off on good footing, users will abandon it before you have the chance to build out more functionality. So let's talk a little bit about the benefits of MVPs. There's a lot of reasons why MVPs are helpful. They allow you to get the product to market fast or much faster than if you built out the entire feature set. By launching an early version of your product, you'll start to get signals if you have product market fit and additionally, you will get lots of feedback from users that can help to inform what you should do next. Lastly, MVPs allow you to fail first and learn. Because this is an iterative approach, you can respond to incoming feedback and course-correct as necessary or in worst-case, abandon the product without having spent a lot of effort and time to build out the full product. So how do you go about creating an MVP? Well, you can use the business model canvas to help define your MVP by specifically looking at customers, the value proposition, channels that you have, and the relationships that you want to build with customers. This will help you to identify what's the right problem to solve and the right value proposition to create for your

customers early on. In order to do this though, you need to understand who are your users and you need to give them a reason to use your product. Once you form the basis of your MVP, you'll want to compare it to other solutions in the market. Is it the same? How is it different? Why would a user choose it over a competing product? You'll want to make sure that you have a compelling answer here. Then you'll also want to make sure that the MVP is aligned with business objectives. Do you have revenue targets? Do you need to launch by a certain date? Are you're trying to attract new users? You need to make sure that your MVP is helping to meet the business objectives, whatever they are. Once you've done all these things, then it's time to break down the MVP into requirements. This is what will make it actionable for the team to execute against, and this is where you're going to start to detail what the product does and doesn't do in more detail. Usually this means writing up the PRD. Again, remember even though it's called minimum viable product, it's still needs to be a complete product that users would be willing to pay for. It's really important to find the right balance in terms of what functionality to build, and how complete the product feels. But keep in mind that an MVP, by definition is only must-haves, and lastly you'll want to identify KPIs. This will help you to understand how your product is doing overall. We'll talk a little bit more about KPIs in the next concept.

So let's talk about KPIs or key performance indicators. KPIs measure the performance of your product. They help you to understand how well your product is working or isn't working as well as how changes that you're making to your product move the needle. There are a couple of different types of KPIs. You can have business KPIs, which measure how you're doing against business goals. You can have product KPIs, which measure how the product is being used and how the product is helping users meet their goals. You can have quality KPIs, which measure how often users encounter issues or problems with the product. You can have development KPIs, which measure how well the team is executing when it comes to building and delivering the product. So how do you go about picking KPIs? Well, don't measure everything. I know it's really tempting to do because there's lots of data and there's lots of things that you can measure, but you want to be super selective when you're picking KPIs. It's better to have a few actionable KPIs rather than a ton of KPI's that are difficult to understand. If you have too many KPI's, you'll waste a lot of time trying to understand what they all mean and what's causing them to change over time. So the fewer metrics that you chase, the more focus that you'll have as a team and the more likely that you will be able to move those specific KPIs. The best KPIs capture user, product and business goals. But it's also okay to have a few KPIs that touch on each of these. If one of your business goals is related to revenue, revenue will likely be one of your KPIs. User goals are related to the problem that your product is trying to solve. Product goals are generally related to the usage of your product, usually around engagement. You also want to make sure that the KPIs that you identify are measurable and that the team agrees on how those KPIs will be measured. At first, this seems really simple and straightforward, but oftentimes as you start digging into the data, there's a lot more complexity than it might appear at first. It's also generally more

helpful to set goals based on achieving things, based on percentages with a certain time-frame when you're talking about KPIs as well. So let's imagine that we're defining KPIs for Zoom just to make this a little bit more tangible. So assuming that the business goals are related to revenue targets, the things that I think would be important to measure are average revenue per user, or ARPU, the conversion rate moving from a free account to a paid account, and the churn rate in terms of how often do users stop using the product. In terms of product KPIs, this is really about how the product is being used. The things that I think would be interesting to consider understanding the weekly active users. So how many users rely on using Zoom at least once a week in order to get their work done and collaborate with their team. Also, how many calls does the average user participate in per week? Another thing that I think it would be important to understand is how long after an account is created does it take for a user to set up their first call with someone else? In terms of quality, this is about understanding how well the product is actually working. There's a couple of things that come to mind. One of the biggest value props of Zoom is that it's high-quality video conferencing delivered in a very simple way. So I would want to make sure that we're monitoring and tracking the percentage of calls that are delivered in HD. Since this is one of the key value props. We also ask users to rate each call after it ends. So that would be another helpful metric and KPI to track here, because this reflects the user's perspective on how the product is working. Another good one to have is the support contact rate. So out of a 100 calls, how many users end up needing to contact support for help? Lastly, for development KPIs, this is about understanding how well the team is executing. I think that it would be helpful to understand what percentage of futures are delivered on time and if there is any slips, what's the average deviation from the original estimate? I also think that it would be helpful to measure how many outages occur per month and on average, how long do those outages last? It's super important to identify KPIs because it gives you a view into how your product is doing at a glance. It's something that your team has also agreed and acknowledged as being important things to track and understand.

So now that you know a little bit more about KPIs, take a little bit of time and think about the KPIs that you would want to measure for the fitness tracker example. Keep in mind the components that you've just mapped out when you were putting together the business model canvas. You'll want to think about how you can translate some of those things into KPIs across business, product, quality, and development dimensions. Then, once you've identified a few KPIs, check back in the next video to see the KPIs that I identified.

Here are the KPIs that I identified for the fitness tracker. For business KPIs, I think it'll be important to track the conversion rate from the free tier to the paid subscription. I think it also will be important to track how often subscribers are churning and canceling their subscription. For product KPIs, the whole value prop is to help users reach their fitness goals. So it will be important to measure the average amount of time that it takes for a user to reach their first goal. The thought here is that if people achieve their first goal more quickly, they're more

likely to continue using the product. Similarly, I think it'll be important to track the average number of sessions per active user. This will give us a sense of how engaged users are and the average number of goals completed per user per month, which gives us a sense of how successful users are being with our product in terms of meeting their fitness goals. For quality, we're looking for a signal from users about how the product is working. I think App Store rating is something that we can easily track and will give us a little bit of visibility into how people are perceiving the app. But I think we can also consider asking users to rate the interactions that they're having with their trainer. Additionally, we can monitor support contact rates as well. Lastly, on the development side, on-time delivery is always a good thing to track and measure in terms of team velocity. Another thing that's usually it really helpful to track is test coverage, high test coverage will allow you to quickly identify and isolate regressions or bugs when they happen. How did these KPIs compare to the ones that you identified? Keep in mind that the value prop for your fitness tracker might be slightly different than mine though.

At this point, you should be able to define and craft compelling vision of what your product is at its very core, as well as build a vision that gets everyone on the team excited. You should also be able to identify strategic areas to invest in across user needs, key features, differentiation, and business goals in order to make your vision and product a reality. You also learned about how you can build a business model canvas in order to map out everything that's required to build your product and the associated costs, the value prop of your product, as well as how to get your product to your targeted customers and generate revenue through a variety of different revenue models. You should also be able to conduct a competitive analysis of competing products, whether those competitors are direct, indirect, or replacement competitors, as well as evaluate the different strategies that those competitors are using, along with the associated pros and cons of each. You should understand the importance of defining an MVP as a vehicle to get to market sooner and validate your product before you invest more time and energy into it. Lastly, you learned how to define KPIs across a number of different dimensions like business, product, quality, and development in order to track and understand if the product and the team are meeting their goals. All right. I will see you in Lesson 4 where we'll start to talk about communication skills.

At the end of the day, PMs get things done by communicating with a number of different people. Communication is an incredibly important skill for a product manager to have and it can take a lot of different forms, whether that's email, in-person, presentations, or even in written documents like PRDs. As a PM, it's super important to be able to effectively communicate your ideas and get people excited about them. In this lesson, we'll cover the following topics. We'll talk about active listening, which is a big part of communication is listening to really understand other perspectives and points of views so that you can respond more effectively. We'll also talk about storytelling, which is a way to simplify your message, make it more engaging on an emotional level, and create something that's more memorable

and easy for people to remember. We'll talk about being persuasive and how you can structure your message in a way that's compelling to your audience. We'll also talk about presentations. As a PM, you'll spend a lot of your time making presentations and we'll walk through some things to keep in mind when you're building slide decks to make sure that your presentation is easy to follow. We'll also talk about negotiation and different strategies that you can employ there. Are you ready? Because I am. Let's get started.

With communication, everything starts with listening. Whether it's the customers, leadership or the team. You need to listen in order to understand other perspectives and needs. Listening can also help to build relationships, promote understanding, and build empathy. It can also help to save time and make sure that things are done right the first time. So what exactly is active listening? Well, it's more than just listening. Listening is really important, but often times communication is not just about what was said, but about how it was said. You have to read between the lines, as well as be aware of body language. When you're actively listening, you're being an engaged listener. This means that you're paying attention to the person who's speaking and making sure that they know that they have your full attention. There shouldn't be any distractions like email or text messages. It's not just about listening. You want to fully understand other points of view. You don't necessarily have to agree with those other points of view, but you do need to understand those views as if you were in the other person's shoes. Remember, you are listening, not imposing your own views at this point. So when you're listening, you're role is not to judge or offer alternative views. It's just to listen and understand exactly where the other person is coming from. So why is active listening important? Well, there's two really big benefits. It helps to build strong relationships. By actively listening, you're building empathy for the other person's point of view and you're also showing that you care about understanding their perspective. This can go a really long way towards building strong relationships. It also promotes understanding. You don't just learn what the other person's point of view is. You understand the context behind why they have that point of view and how they got there. This gives you a much deeper level of understanding. As well, it can help you to anticipate how they might react to certain things in the future. So let's walk through some tips for how you can actively listen. The first thing you'll want to think about is making eye contact and smiling. You want the other person to feel relaxed and comfortable when they're talking to you. You can help to promote this by making eye contact and smiling when appropriate. This shows the other person that you're listening to what they're saying. Another thing that you can do is provide encouragement. So what this means is that you're acknowledging that you're hearing what the other person is saying. You can do this by nodding your head or making small comments like saying, "Yes," "mmhm," or "that makes sense." Another thing that's important is don't interrupt. Give the other person the space to complete their thoughts. You want to avoid interrupting them. Let them finish what they're saying and make sure that you hear and fully understand what they're saying. But you can ask clarifying questions when needed. If you aren't quite following what they're saying, wait until there's a pause and then ask them if they can clarify. Raise your questions

in a way that avoids judgment. The best types of questions are open-ended, where you genuinely want to better understand their point of view. Lastly is paraphrase. It's also really helpful to paraphrase and mirror back what you just heard. This makes sure that you're really understanding what the person is saying and gives them an opportunity to correct you if you misheard. You can do this by saying something like, "Just to make sure I'm understanding this, I think that you just said XYZ. Did I get that right?"

Now it's time to practice your active listening skills. Identify two friends or colleagues that you could have a short 15 minute conversation with. Pick a topic and really, really try to understand their point of view on it. For example, you could ask them about a product they use. Why do they like it, or not like it, and how do they think it could be better? Or you could ask them about what they did yesterday and why they did those things. Alternatively, if you're having some trouble thinking of topics, you can use these questions about goals as a guide. You could ask your friend, what are your goals over the next six months, why are these goals important to you, what are some of the challenges that you might encounter, and what would help you reach these goals faster? So go ahead and give it a try with a couple of friends.

Now we're going to talk about storytelling. At the end of the day, PMs are storytellers. What exactly is storytelling? Well, it's a powerful tool that you can use to create more tangible narratives. Storytelling is a tool that allows you to express your vision and thoughts through more compelling, tangible narratives, and it makes it easier for people to understand and relate to what you're explaining. Storytelling explains what and why. This is because the story explains not just what happens, but why it's happening as well. Stories explain the rationale behind what's going on. This provides the audience with deeper insight and understanding around motives. Stories also have an emotional element. When you tell a story, the audience builds an attachment to characters in that story. This creates more of an emotional response and gets the audience invested in the outcome of the story. Stories can also be exciting, and when you get people excited about solving the problem, they're really going to care about solving the problem and be invested in it. Lastly, storytelling is a core role of being a PM. You'll need to tell stories all the time, whether it's explaining your roadmap to leadership and the larger company or building user stories for the design and engineering team to execute against. Stories are a great way to inspire the team and get everyone excited. So now let's talk a little bit about the elements that go into a story. First, you have the hero. This is your user, who they are, what they care about, their background, and then there's a goal. This is what the hero is trying to accomplish. But before they can achieve their goal, they encounter a villain or a problem that's preventing the hero from getting to where they're trying to go, and that's where the sidekick comes in to help. The sidekick in most cases is going to be your product. Then lastly, you have the plot, which describes how the hero and the sidekick defeat the villain and achieve their goal. So what else goes into crafting a story? Well, you're going to want to start with the big picture. It's important to start with the big picture and provide context. You'll want to make sure that you understand who your audience is and what they

care about. You'll also want to think about how you want them to feel at the end of the story and what actions you want them to take if any. With that in mind, then you'll want to describe the problem or the conflict. It's usually helpful to start by explaining how we got here and why did this problem arise in the first place. You can also touch on why this is an important problem to solve and what's currently preventing your user from reaching their goal. Then comes to the plot for how to get to resolution. You want to make sure to include enough imagery and detail so that the audience gets a mental image in their mind. The more personal you can make it, the better. How does your product help the user to reach their goal? How is that any better than other options that are available? Then lastly, you'll want to talk about the impact and the conclusion. How was the problem ultimately solved? Why does it matter that it was solved? What does it mean for the user? So now I'm going to share a few tips that you should keep in mind when you're crafting a story. First and foremost, you want to keep it simple. The simpler your message is, the more that it will get across to your audience. After you've created your story, one of the things that you can do is you can ask yourself how you would tell the same story, and just half of the time. That's a really great way to go through and ask yourself and find out if you actually need every single element that you've created so far. You also want to make your story memorable. You want people to be able to remember the important parts of your story, and a really great way to do this is by making it more personal and relatable, and by having emotional moments that people might have experienced themselves. Another thing that you can do with a story is you can show why things matter rather than telling or explaining. You don't want to explain what your product does in detail, but you can show what it does and you can show its impact. You can show how people would use it and why they're using it. Potentially, this is also a point where you can include mocks or demos of your product as well. Another thing that's super important is practice: practice, practice, practice. You should be able to tell the story without having to refer to any notes. This doesn't mean that you've memorized it word by word though. So we just talked a little bit about storytelling, what it is, and why it matters, and how you can create stories as well.

So now I'm going to walk through two examples of storytelling. First, I'm going to start with a weak example. Today, people are busier than ever and they're constantly on the go. We have an opportunity to build an app that allows customers to order ahead. People will be able to place their orders ahead of time and the order will be ready waiting for them when they show up. This is super convenient because they will be able to just walk in and grab their order on the way to their destination during their busy day. Now for a stronger example, we're going to make the story a little bit more personal. John is a coffee person, and he just really isn't awake in the morning until he's had a cup of coffee. But the morning is usually a pretty chaotic time for John. Between waking up, getting ready for work, walking the dog, and getting into the office early for conference calls with a team that's a few time zones ahead. This means that John usually doesn't have time for coffee in the morning. The first few meetings of the day end up being a little bit rough, as a result. But John recently heard about

a new app that lets you order ahead. So one morning John decided to try it out and placed in order as he was leaving his house for the office. He was able to pick it up at the store that's right across the street from his office. He just walked into the store and his order of coffee was there waiting for him. So he made it into the office with his coffee in hand and jumped right into his first meeting of the day, energized and ready to go. So those two stories explained very, very similar things, but the second story was much more interesting, and much more memorable.

Now it's time for you to practice telling a story. I know this sounds really cheesy, but it's super helpful to practice rehearsing your story out loud in front of a mirror. Then once you've created your story and practice it, it's time to share it with a friend. After you share the story, ask your friend to summarize your story. You'll also want to ask them, how the story made them feel, and describe what they thought was the most important takeaway. After your friend has answered those questions, think about how their responses compared to what you were expecting them to say, and if they were different, why do you think they were different?

So now we're going to talk a little bit about how you can create persuasive arguments. But before we do, I'm going to give you a quick overview of why it's important to be able to persuade others. So why does persuasion matter? Well, for a product manager, it actually takes a handful of people to build a product, and you work with all of them. But in order to build a successful product, the team needs to be aligned and moving together as a group in the same direction. But you don't have direct authority over everyone that you work with. So you can't just tell them what they need to do or believe. Instead, you have to convince them. You have to persuade them. So let's dig into persuasion a little bit deeper now. There are three different modes of persuasion according to Aristotle, a Greek philosopher. There's ethos, which is your credibility as an individual. This happens when you're qualified to speak on a topic, or after you've spent time working with the team, you will earn and gain the trust of your team. But this trust is also something that can be achieved through your personal character, background, and other achievements. There's also pathos, which is emotion and making your audience feel a certain way. Then lastly there's logos, which is logic, and this is building an argument based on fact and data. These are truths that can't be refuted. The best messages have all three. It's important to think of these types of persuasion as interconnected rather than stand-alone. For example, having strong logic and data will increase your credibility, and using certain pieces of data might actually provoke an emotional response from your audience. You also can use well-placed humor to help you connect with your audience and gain more credibility. So how can you build persuasive messages? Well, first, you need to know your audience. It's super helpful to understand who your audience is and what they care about. This allows you to tailor your message to specifically address those concerns. Next, you want to pick a viewpoint. Be decisive. Pick a viewpoint and explain why it's the best outcome. Too often I've seen people try to explain all of the different options. This ends up diluting your message and creating confusion around what it is you're actually

recommending. But having said that, you still will want to know all of the other perspectives and be ready to speak to them if they come up, including the associated pros and cons of each. You'll also want to be data-driven. Support your claims with data. You don't need to go over the top, but have a few key data points that can really help you to drive your message home. Another thing that's really important is to discuss objections. When objections or questions come up, don't try and dismiss them. This will actually make you seem defensive. Instead, have a real discussion around the merits of those other viewpoints and try to get the group to a consensus on them. This shows the audience that you care about their viewpoint and that you're also invested in getting to the best possible solution, whether or not it's your own idea. Lastly, you'll want to end with a call to action. So you just finished your presentation, and you got the entire group aligned with your point of view. Make sure to wrap it up by outlining the next steps so that everyone understands what was agreed upon and what will happen next. Remember, you're not going to win every discussion and that's okay. But over time, what will happen is you'll build confidence in yourself and how to best prepare, and the people that you work with will also build confidence in your ability to deliver.

Now we're going to talk about presentations. As a PM, you will need to be constantly communicating. Presentations are a great tool that you can use to share your ideas. But before we get into the details, one of the most important things to keep in mind when creating a presentation is to keep it simple. Don't try to cover everything. Just focus on getting your key points and messages across. So now let's talk a little bit about the right structure for your presentation. There's three main parts. There's an introduction. This is what I'm going to tell you and in this section, you want to provide any background information or contexts that will be helpful for the audience to understand your presentation. You also want to get people's attention here. So try to open with something memorable or interesting. Potentially, this could be telling a story. Then you have the body and this is really the bulk of your presentation. This is where you get into all the details. Remember, you don't need to go super deep, but think about what is necessary information in order to get to the outcome that you want versus what is additional healthful context that could go somewhere else, like in an appendix or another document and then lastly, you have the conclusion. This is where you wrap it all up and you tell everyone what you just told them. You also want to bring up any next steps at this point. One thing that you want to keep in mind as you're working through the outline of your presentation is that you have a time limit and people have the tendency to ask questions. So you usually have less time than the overall time for the meeting. So keep that in mind. So let's talk a little bit about slide layout. One of the most important things is to keep your slides simple. I know it can be really tempting, but you want to avoid using lots of loud colors or colors that conflict. You want the slides to be easy to read. You also will want to minimize the use of transitions unless it's something that's adding a lot of value to the meaning of the message that you're trying to explain. So here's an example of a slide that has a lot of transitions, a lot of really bright and loud colors, and a lot of different font styles as well. None of these things are actually helping to get across the meaning that you are trying to convey.

Here's a better example. There's a lot less going on on this slide, but that's actually okay, because it's much easier and quicker to understand all of the key points. Another thing related to keeping your slides simple is that you want to have one key takeaway per slide. This is because people will start reading the slide as soon as they see it. So you want to keep all of the context focused on the key point that you want to get across and then once you've covered everything, then you can move on to your next key point. So here's another example of the slide. Again, there's a lot going on on this slide. There's a whole bunch of data, but I don't really understand its significance. There's no one interpretation at all. So I'm not really sure what the takeaway or the key point of this slide is. Contrasted with this slide, which is much better. I immediately know that team one scored the most points with a 113 points. The slide has the same data as the previous slide. It's just been interpreted and presented in a way that makes it easier for the audience to understand. In addition to highlighting that team one got the most points, I also can quickly tell that team four was pretty close behind. Team two had half as many points as Team one and Team three didn't do very well at all. Another thing to keep in mind when you're working on your sides is you really want to limit the amount of text on each slide. Try really hard to avoid paragraphs. It's a lot of texts and it's really difficult to read. You're not writing an essay here. Your audience should be able to look at the slide and understand its significance in a few seconds. Bullet points are much better at achieving this and alternatively, you can simplify even further and use imagery to represent each bullet point, if that will add value to your message. Visuals can also be really powerful, but they can also cause a lot of confusion. So you need to be very intentional about when to use them and how. Here's a slide that has a lot of Clip Art and visuals going on. The problem here is I don't really understand what it all means. Visual should enhance your message, but shouldn't be the only way that your message gets across. This slide is a much better example. There's a title that explains the key point backed with a photo that makes it super easy to understand what the experience could actually look like.

There's a couple other tips that I want to share for you to keep in mind while you're working on presentations. First, it's important to realize that presentations need to stand on their own. This is because, you'll often send out your presentations. People will read them without you there to talk them through it. So you need to make sure that your presentation has enough detail and information that they can interpret your messages and slides in the way that you intend them to it. It's also really helpful to rehearse and practice giving your presentation. While you're doing this, you can also edit and refine your slides to simplify your message as needed. It also can be really helpful to pre-brief stakeholders for big presentations. This means giving them a preview or running through your presentation with them one on one. This allows you to get their feedback before you give their presentation, so that you can take it into account instead of having to address those questions in front of a busy room during the meeting.

Now it's your turn to put together a short presentation. Keep in mind all of the best practices

and tips that we just walked through. Your presentation should answer the question why should you avoid using your smart phone before bed? You'll want to make sure that your presentation is at least 10 slides long and that you've clearly identified the key takeaways that you want your audience to walk away with. Once you've put together your slide deck, the next thing you'll do is present it to a friend. After the presentation, ask them for feedback. Specifically, there's two areas that will be important to get their thoughts on. The first is understanding if there were any parts that they thought were confusing, and why. Secondly, you'll also want to understand what they thought was the most important takeaway. Try not to overthink this too much. I think it's something that you can actually have a lot of fun with.

Now let's talk a little bit about negotiation, which is an important skill that PMs use every single day. Negotiation is about trying to reach an agreement when there's conflict. Conflict can arise in lots of different scenarios, whether it's about what the product should do, who's going to be working on the product or the specific timelines of what's happening when. Negotiation also involves compromise, which is about finding the best mutual outcome for all parties involved. When you're negotiating people tend to think of it as either being a win or lose situation. But there's actually a couple of different styles of negotiation. There's competitive negotiation, where I'm going to win and you're going to lose. There's collaborative negotiation, where you and I both walk away feeling like we won. There's compromise, where we both walk away feeling like we won and lost some parts of the negotiation. There's avoidance, where we both lose, and this usually happens when people don't like conflict, and so as a result, the issues end up not being directly discussed. Lastly, there is accommodation. This is where I lose and you win. This isn't necessarily in my best interests, but sometimes it might be needed in order to preserve relationships. These styles will all come into play at different times. But most of the time when you're working with your team and within your company, collaborative negotiation is going to be the best approach. So how do you get there? What are different negotiation strategies? Well, the first thing that's helpful to understand is BATNA, which stands for Best Alternative To a Negotiated Agreement. You'll want to understand what each side's BATNA is. Think about what would happen if you weren't able to reach an agreement? What would the outcome be for each party? Another thing that's important is to focus on interests. It's really helpful to understand what each side is interested in and why? These insights will help you to understand where to focus during the rest of the negotiation. Another really great strategy is to make the pie bigger. What this means is you'll want to look for ways that you can increase the value of the negotiation. You can do this by understanding the other side's interests. Are there any things that you could offer them that will be valuable for them? In the best case, those things won't cost you anything or be difficult for you to give away. Another thing that you can do is use objective criteria. By creating an objective criteria, it helps to base the conversation in reality. Imagine, for example, that you're interested in buying a car. The salesperson wants to sell the car to you for as much money as possible, and you want to buy the car for as little money as possible. Using objective criteria helps to ground the conversation, because you're able to

look at similar models with similar mileage and see and understand what those cars are selling for. It can also be helpful to present multiple solutions. When you do this, you'll get feedback about how the other side feels about each option and which option they think is the best. So let's walk through a few tips for negotiation. First of all, relationships matter a lot. When you build strong relationships with people, it's easier to negotiate because each side has a certain level of trust in each other. Compare this to two people who have never interacted before. You can also use active listening. By using active listening, you'll understand the other side's point of view and why they have that point of view. Again, the better you can understand their motivations, the easier it will be to identify a solution. Active listening also involves paying attention to body language to gauge how the other side is responding to the conversation. It's also really important to keep your emotions in check. Negotiation can be stressful and it's sometimes very emotional. It's important to keep your emotions in check and stay calm and relaxed. If your emotions run unchecked, things can escalate quickly and can lead to bad outcomes. Another tip is to work together to find a solution. You can do this by involving the other side and finding a solution. A really good way to do this is to start by aligning on goals. If you can both agree that you're working towards the same goal, it's way easier to work together to figure out how to get there.

Let's go through a quick recap of what we talked about in this lesson. At this point, you should be able to use active listening to better understand other people's points of view. You should also be able to craft and deliver compelling stories to get your message across. These are stories that should be exciting and memorable. You also learn how to create persuasive messages using three different modes of logic, as well as being able to create easy to understand presentations using best practices. We also talked about how you can negotiate to optimize for win-win situations. I wanted to take a few minutes to recap what we covered throughout the entire course. You should be able to describe the role of a PM and a PM's critical partners, build roadmaps and write PRDs, identify and size opportunities, create a business case for a new product opportunity, define the use cases of a product including its KPIs, build a business model for your new product including understanding different types of revenue models, and put together and present a compelling product pitch in order to gain internal stakeholder buy-in. Congrats again for making it through all four lessons. We covered a lot of topics related to identifying and evaluating opportunities for new products. I'm looking forward to seeing you in the next course where we'll take those opportunities and start turning them into more concrete ideas.