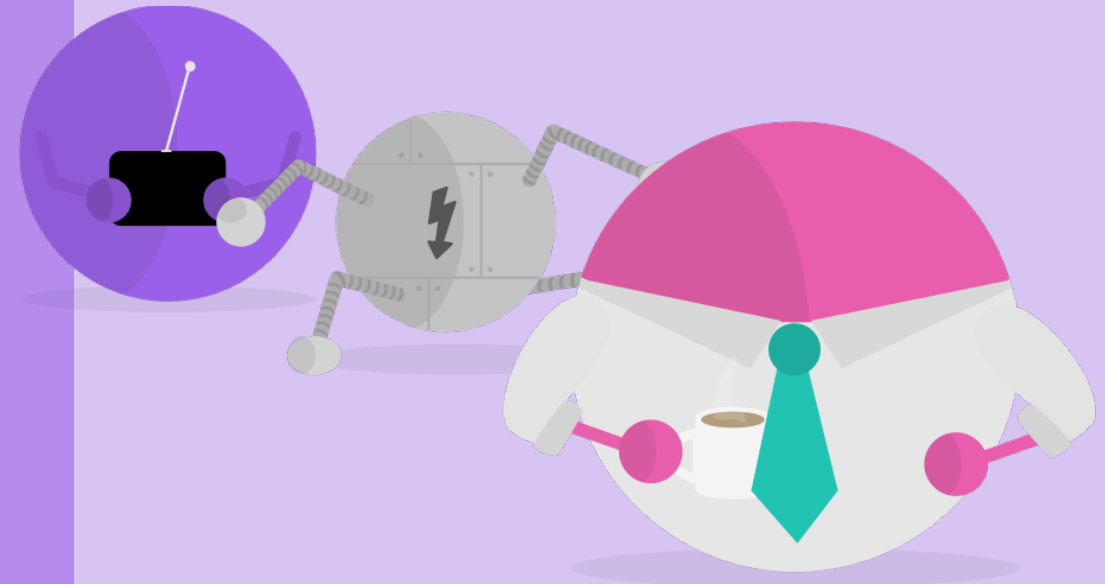




The Product Manager's Guide to Building AI Products

Everything you need to know when creating and managing AI features and functionality



Contents

Introduction	3
General considerations when building AI products (or AI into your product).....	5
What are your motivations for adding AI?.....	6
Do you understand the 'magic valley'?	6
Have you set realistic expectations?	7
Are you ready to move fast?	7
Do you know the risks?	7
Is the rest of your organization ready for AI?.....	8
Do you understand the ethical and legal requirements of AI?	8
Are you willing to be boring?.....	8
Building a new AI product from scratch.....	9
What are the challenges of building an AI tool from scratch?	11

When is building a new AI tool from scratch a good idea?	12
Adding AI into an existing product.....	13
Why add AI into your product?	14
Considerations when adding AI into an existing product	15
Using an existing model vs building your own	17
Building your own AI model.....	18
Using an existing AI model	23
How do you know which AI model to use?.....	25
Considerations when using an existing AI model.....	26
Which way will you go?.....	28
Pricing AI	29
How to manage an AI product.....	29
Challenges of AI for Product Managers	38
Conclusion	42

Introduction

AI has changed everything. It's become a core feature in almost every industry, and we're willing to bet it's now embedded itself as part of your daily routine.

According to research by [Lenny Rachitsky](#), over 90% of product people use language learning models (LLMs) like ChatGPT every single day. That is a hell of a lot, and that's not yet factoring in all the other [AI tools for Product Managers](#) at your disposal.

AI is transforming Product Management in two fundamental ways:

- 1 As an enabler in the products we build and manage.
- 2 As a tool to enhance how we do our jobs as Product Managers.

This ebook is all about the first - how to successfully integrate AI into your product. If you're interested in how to use AI to supercharge your own workflow as a PM, we got that covered too.

Prompt Engineering for Product Managers: How to Get Things Right With Generative AI

[Read now](#)



So, how do you get the most out of the capabilities of AI technology and incorporate them within your products? Well, when it comes to incorporating AI into your product, you'll likely find yourself choosing one of four scenarios:

1. Enhancing an existing product using an existing AI model
2. Enhancing an existing product and building your own AI model
3. Building a brand-new AI product using an existing AI model
4. Building a brand-new AI product and creating your own AI model

We'll talk you through all of these approaches, the skills needed, and the challenges you have to overcome in order to embed AI effectively.

AI is no longer just a niche specialism for a handful of "[AI Product Managers](#)"; it's becoming a core part of the PM toolkit. Soon, **every** Product Manager will be expected to understand how AI fits into their roadmap, their strategy, and their user experience.

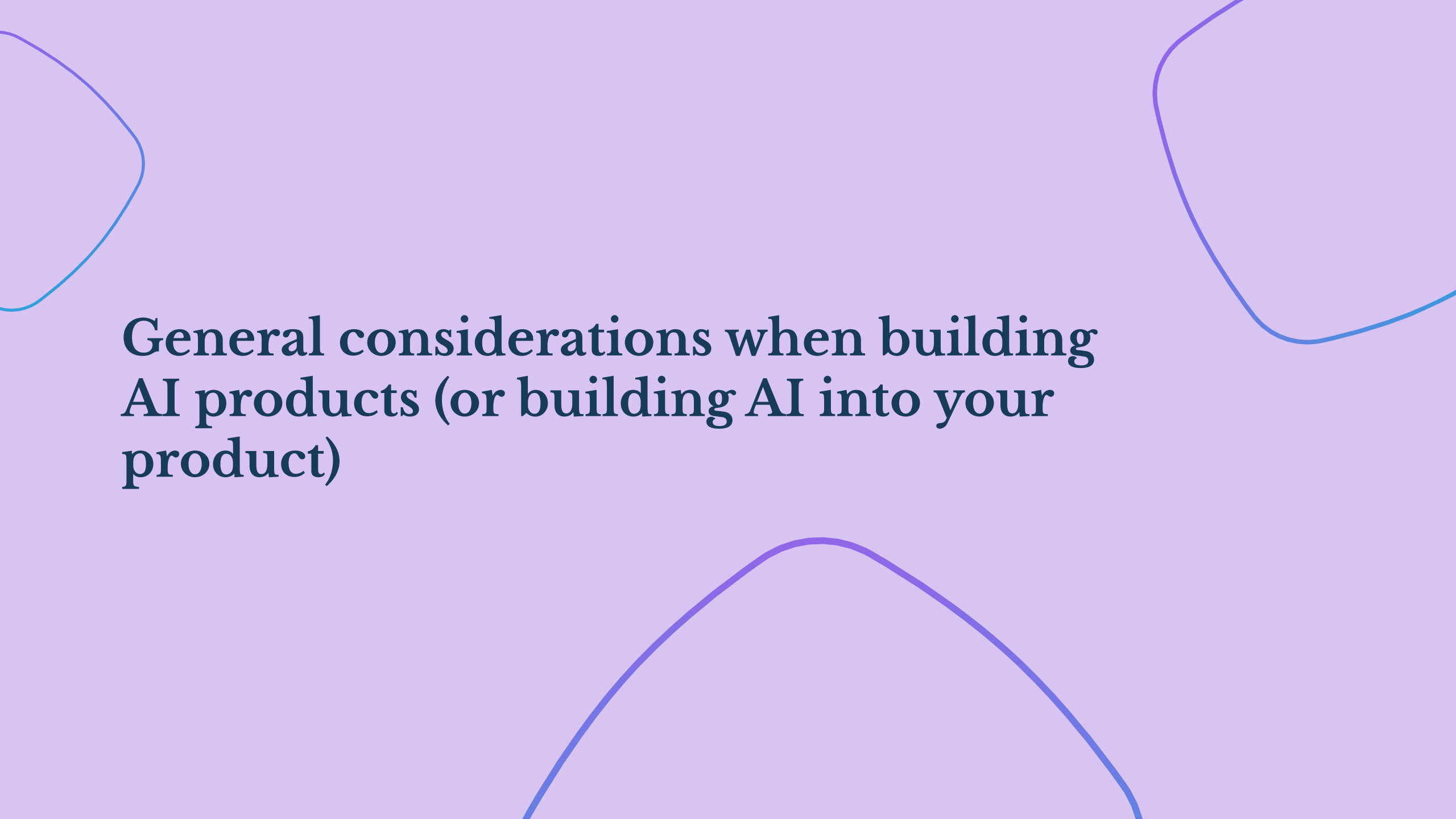
At ProdPad, we know this first-hand. We've been using AI in our own products since 2018, and have built one of the most advanced AI applications for Product Management on the market: **ProdPad CoPilot** ✨.

We've learned a lot along the way, and in this guide, we'll share what you need to know to successfully manage AI-powered products, whether you're just getting started or already deep in the trenches.

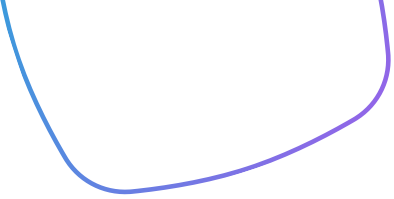
Let's dive in.

Learn more about ProdPad CoPilot, the world's
best Product Management AI tool

[Find out more](#)



General considerations when building AI products (or building AI into your product)



So you want to start offering an AI product? Nice. Now, despite pretty much everyone in tech doing the same, blindly throwing in AI functionality isn't something you should do lightly.

Implementing AI will fundamentally change your product, forever. So before you dive in and dip your toes into the world of AI, there are multiple things you need to ask yourself. Here are some questions to ask:

1. What are your motivations for adding AI?

AI isn't a silver bullet. Adding it just because it's trendy won't add value. Avoid the temptation of **shiny object syndrome**. Ensure AI addresses real user pain points and focuses on solving problems rather than just jumping on the bandwagon. Stick to core Product Management principles like user research and hypothesis testing to decide if AI really is the right solution.

Avoid gimmicks and only use AI in your product if it helps you solve problems for your customers and/or your business.

2. Do you understand the 'magic valley'?

AI can seem like magic, a rabbit pulled out of a hat, but too much automation and mystery can create negative experiences. You don't want to create a tool that weirds everyone out.

The magic valley is an AI equivalent to the uncanny valley in robotics, which describes when people start to freak out and have a negative reaction when a robot's likeness to a human becomes too great. Just as humans have this uncanny valley in robotics and human representations, there is a similar valley for the "magic" of AI.

So if your AI tool does too much without much input, you'll hit the magic valley and put users off.

AI without the right level of control or user input can be unsettling and untrustworthy. But too much control and required input makes the tools useless and removes the USP of AI. You need to manage this "magic valley" and find the sweet spot between too little and too much automation.

3. Have you set realistic expectations?

AI isn't perfect. It makes mistakes. It hallucinates. And no, it won't single-handedly revolutionize your company overnight.

Too many AI-powered products market themselves as world-changing, only for users to be short-changed when reality doesn't match the hype. Instead of selling AI as a flawless solution, be transparent about what it can - and can't - do.

Make sure your customers are clear about what your tool can actually do, but more importantly, make sure your internal stakeholders are clear on it too. You don't want to bet everything on an AI tool and expect it to solve all your problems. Your expectations need to be based on real-world, achievable results.

4. Are you ready to move fast?

AI is one of the fastest-growing sectors in tech. It evolves rapidly. Models improve, APIs change, and competitors iterate quickly. If you're using AI in your product, you need a strategy that allows for **continuous improvement**.

This means being prepared to pivot and adapt quickly as new opportunities arise or as your AI models and features evolve. Flexibility will be crucial in staying ahead of the curve and ensuring your AI-driven product stays relevant.

5. Do you know the risks?

AI introduces new risks across security, reliability, and trust. Are you across them all? Some key considerations include:

- **Data security:** What customer data is your AI processing? Could sensitive data leak into the model?
- **Non-deterministic behavior:** AI isn't deterministic - meaning the same input won't always produce the same output. Can your product handle that unpredictability?
- **Bias and fairness:** AI models can reflect and amplify biases. How will you ensure fair, ethical decision-making?
- **Compliance and regulations:** AI laws are still evolving, but GDPR, CCPA, and AI Act-like policies will impact your offering. Are you prepared?

We'll cover all the risks in more detail in our chapter on the **Challenges of AI**.

6. Is the rest of your organization ready for AI?

Introducing AI is an organizational shift, not just a product or engineering challenge. If your company isn't prepared to support, sell, and market it, your AI launch could slip at the starting line.

Involve key teams early on to ensure they're ready for the changes ahead. For example, Customer Success needs to understand how to help users adopt AI, while Legal must ensure compliance.

Sales and Marketing also need time to grasp AI's value, adjust their messaging and demos, and be prepared to answer technical questions.

Aligning all teams early is crucial for a smooth, unified launch.

7. Do you understand the ethical and legal requirements of AI?

The legal landscape around AI is still shifting. Depending on your product, you may face different levels of regulatory scrutiny.

Sure, AI-generated user stories have few ethical concerns, but if your tool is in the realm of finance or recruitment and is powering things like credit approvals or hiring decisions, then there are huge ethical considerations there.

If you're offering AI in a B2B product, enterprise customers will ask about security and compliance. Have your answers ready.

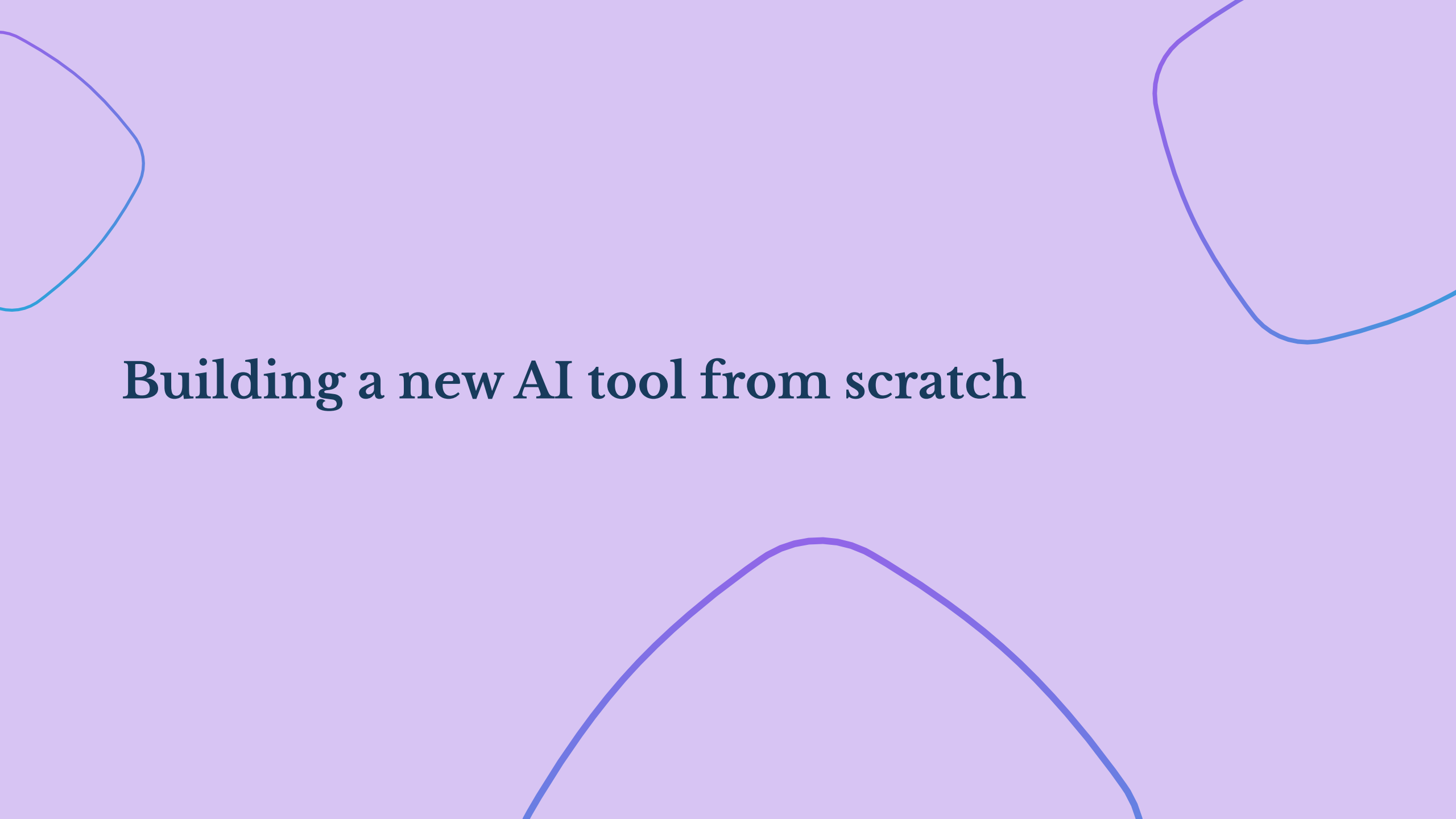
8. Are you willing to be boring?

AI is exciting. It can generate stunning images, code entire applications, and do things that once seemed impossible. But the most successful AI applications? They're often... boring.

The biggest wins for AI-powered products aren't in sci-fi-level innovations but in eliminating tedious, repetitive work.

Take our **CoPilot AI**. It can do a lot, but do you want to know one of the most common uses with our users? Generating user stories. Why? Because it removes a mundane, time-consuming task, freeing up Product Managers to focus on strategy and decision-making.

The best AI features don't just impress users, they make their lives easier. If you're only chasing the "cool factor" of AI, you might miss the real opportunities.



Building a new AI tool from scratch

Made your considerations and found that AI still is the best idea for your product? Awesome. Well, from here, you have two main routes to go:

Add AI functionality to your existing product or create a new, separate AI-only product from scratch.

We'll be honest with you right off the bat, that first option is BY FAR the most common approach when it comes to introducing an AI tool, but there are scenarios where crafting a standalone AI product from the ground up is better suited.

While building an AI tool from scratch might sound like an exciting opportunity to create something innovative, it's also one of the most challenging.

See, a lot of AI-driven products succeed because they leverage existing user bases and data sources - two things that a brand new AI tool won't have at launch. That doesn't mean it's impossible, but it does mean you need a clear strategy to make it work.

Of course, one of the main things you need to be sure of before creating an AI tool is that there's actually a demand for a standalone AI product. You need to do your **product discovery** and validation and identify that there is a problem that's being solved by your new AI tool.

You can't just decide to create an AI-native tool just because the technology is popular and it's the expected thing to do. It needs to be the right thing for your company and market.

We'll get into what the ideal scenario for building an AI tool from scratch looks like, but first, here's a breakdown of the challenges associated with this approach to introducing AI. Just so you know what you're dealing with.

What are the challenges of building an AI tool from scratch?

❌ **Lack of data:** One of the most significant hurdles of building an AI product from scratch is the lack of data. AI relies on data to function effectively, and platforms that already exist in the market have the advantage of years of user data. Without that data, you'll need to find ways to gather the context your AI needs to provide valuable insights.

❌ **No inbuilt customer base:** One of the biggest challenges of building an AI product from scratch is that you're starting with zero users. Unlike adding AI to an existing product - where you can immediately offer value to an established audience - you'll need to attract customers from the ground up. That's never guaranteed.

❌ **Trust & credibility issues:** New AI products must overcome skepticism. Unlike established platforms, which have established trust, a new AI tool must prove its reliability, accuracy, and security from day one. Convincing users to trust your AI with their data or decision-making is a significant hurdle.

When is building a new AI tool from scratch a good idea?

If you're ready to take on those challenges and build and launch a new AI product onto the market, there are a few factors that could give you an advantage. If you think any of these apply to you, chances are you're in a strong position to make a go of this.

You have proprietary data

AI is only as good as the data it's trained on. If you have access to a unique dataset that no one else can replicate, whether it's industry-specific insights, user-generated data, or proprietary research, you have a competitive edge. In this case, building your own AI tool allows you to create a unique offering that should outperform generic AI solutions.

For example:

- A **healthcare company** with exclusive patient outcome data could build an AI-powered diagnostic assistant that provides more accurate recommendations than off-the-shelf solutions.
- A **financial services** firm with decades of transaction history might develop an AI-powered risk assessment tool tailored to its client's needs.

The problem is underserved

Broad AI tools like ChatGPT or Bard are powerful, but they're designed for general use cases. If your target audience has a unique workflow or industry-specific needs that aren't being addressed by existing AI-powered platforms, building a new AI tool could be the right strategy.

For example, legal professionals need AI that understands case law, citations, and legal jargon - something generic AI models may not handle well. A startup focusing on AI-driven legal research could justify building a specialized tool from scratch to serve this specific market need.

If existing products fall short in critical ways, a focused AI tool built specifically for your audience may provide far more value.

But what if you're currently managing a long-standing, or even short-standing product, and you want to enhance your offering with AI? Are the considerations the same?

Short answer: no. Let's cover that in the next chapter...



Adding AI into an existing product

If you're not building and launching a brand new AI product, chances are you're adding AI technology into the existing product you manage. After all, adding AI capabilities into your product can seriously help you offer more value to your customers, increase your competitive advantage and power your growth.

You're arguably in a better position if you have an existing product to enrich with AI. You already have data, a customer base and market presence. But just because you CAN add AI to your product, does that mean you should?

Why add AI into your product?

This is an important question, as you risk a number of pitfalls if your motivation for doing this isn't right. There are quite a few solid reasons for adding AI into your existing product. It's undeniable that AI technology can help you offer more value and enhance your product considerably - if done right.

Adding AI into your product can be a great way to help you:

- ✓ Gain competitive advantage
- ✓ Better solve the problems your product is currently focused on
- ✓ Solve new, related problems that your customer base would value
- ✓ Open up the use cases of your product by solving problems for a new set of users

But, it's worth taking a step back and really considering your motivations. We touched on this in an earlier chapter, but it's worth saying again. **Don't add AI to your product for the sake of adding AI to your product.**

As with any product feature, it's only going to be valuable to your business and your customers if it solves a problem. So when considering adding AI to your product, focus on the outcome you want to achieve, the value you want to provide your customers, and explore how AI can help you offer a solution. Avoid gimmicks and focus on solving problems.

You can move fast!

There's a major advantage to adding AI to an existing product versus starting from scratch. With an existing product, you already have a database and structured data. This is the food that feeds AI - and you have it from the get-go.

Your existing data means your AI features can immediately start doing cool stuff for people. You can use everything that already exists in your product - be that customer information, their context, their work, their preferences - and provide that context to the model and get truly useful and relevant outputs.

Considerations when adding AI into an existing product

Adding AI to your product is useful, but there is still a good list of unique things you need to think about when it comes to integrating this technology.

How is your data structured?

Although you can enjoy the advantage of already having data to feed the model, from your existing customer base and their activity, information, work (whatever it is your product stores for them), there might be an issue if there are parts of your application where the data isn't structured in the right way.

If the data is unstructured or structured in such a way that AI can't work with it, it could actually get in the way of AI and sabotage your efforts.

Are your customers ready?

When adding AI to an existing product, you'll be bringing your current customers along for the ride. So you need to think about whether they're ready to embrace the AI capabilities you're about to offer.

You could encounter sensitivities around the use of AI and your users fearing for the future of their roles (especially if your tool completes a lot of the jobs they have traditionally done manually).

You might also face security concerns or explicit internal policy stipulations (if B2B) that might hamper their use of your AI technology.

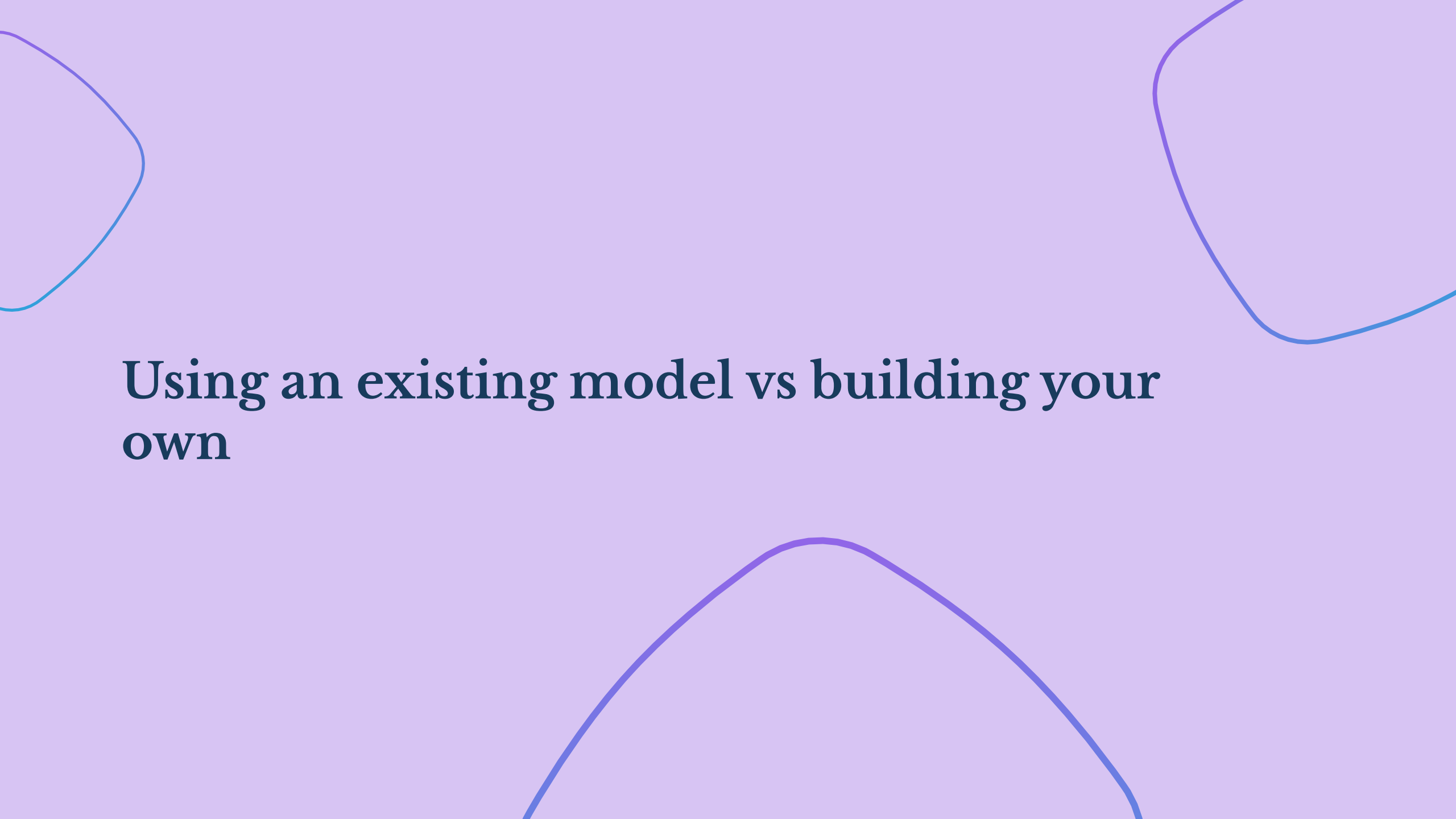
So try and get ahead of these potential concerns and work out what you can do to allay fears, comply with policies and encourage adoption.

How will the AI sit with your current pricing?

Since you have an existing product, you'll have an existing pricing strategy and a pricing model that your customers are subscribed to and used to. You need to consider how the introduction of AI functionality impacts that.

It's best to think about this early on. No matter whether you opt to build your own AI model or use an existing one (see the next chapter for more on that!), there will be considerable costs involved and you'll need to consider how you cover those costs to make this AI venture viable to the business.

Luckily we've covered **AI pricing** in depth in its very own chapter - so read on!

The background is a solid light purple color. It features several abstract, wavy lines in a slightly darker shade of purple. One line starts at the top left, curves down and then up towards the center. Another line starts at the top right, curves down and then up towards the center. A third line starts at the bottom center, curves up and then down towards the right side. These lines create a sense of movement and depth.

**Using an existing model vs building your
own**

Whether you're currently, or planning to, build an AI native product from scratch, or enrich an existing product with AI capabilities, one of the first decisions you'll need to make is how you power that AI functionality.

Where will the artificial intelligence come from? That all comes down to the AI model - do you build a brand new AI model yourself, or do you incorporate an existing AI model into your product?

How do you decide which direction is right for you?

Let's look at each option in turn and outline the considerations you should be aware of.

Building your own AI model

The first thing to say here is that most Product Teams do not opt for this path. Building your own AI model is a significant undertaking, one that is only really worth the investment in a few, fairly rare cases.

With so many open source LLMs available, it's a far quicker and easier process to adopt an existing model and fine tune

it to suit your particular use case. Therefore, your decision to embark on the journey of creating your own AI model should only be taken when you're certain an existing model will not serve your needs.

So when is it right to disregard existing open source LLMs and build your own model?

When should you build your own model?

This usually comes down to how specialized your product needs to be. Are you serving a very particular and niche use case? Is the intelligence needed very specific and not achievable through fine tuning an existing model?

One example of this kind of hyper-specialized product that could require a unique model might be a tool that identifies plant or animal species from photos. In this case, you would need to train the model on a set of very exact images.

Another factor that could contribute to a decision to build your own model would be if you needed the model trained on very specific, proprietary data - data that you know an open source LLM will not have had access to.

It might not be enough to provide an existing model with access to your priority data, since that won't influence the model itself, but only enable the AI to draw on content as part of its output. If you need the model to actually be trained using your data - in other words, having that data influence how the model learns - then building your own model or custom machine learning algorithm will be necessary.

Risk level and potential impact should also influence your decision to build your own model vs using an existing one. Do you need unwavering confidence that your model has been trained on the right data, because a wrong move could have significant consequences?

For example, if your product needs to recognize cancers from a radiography scan image, would you be confident that an existing model had been trained sufficiently to make an accurate assessment? Clearly, in this instance, an inaccurate answer from the AI could be lifethreatening. Here you would be right to consider building your own model and training it on an in-depth set of images.

Considerations when building your own model

If you have read the previous section and feel that your product DOES fall into any of those cases where building your own AI model is right, then here's what else you need to consider before you launch into this major technical endeavor.

Have you got enough training data?

Well, have you? You need A LOT of training data to build a viable AI product. And then, once you've built the model and trained it up, you're going to need some data left over with which to test the model.

It's also not enough to have a mound of data, it also needs to be labeled effectively. It's no good throwing a massive data set at a model and hoping for the best. You need to guide the model and help it understand what it's actually looking at.

Take our example from earlier - a product that assesses radiography scans and spots cancers. We've already discussed the advantages of building your own model for this and training it with a vast series of different images. But this will only work if each image is labeled, telling the model whether it's an image with cancer present or not.

That example would require a massive quantity of images fed into the model - both of scans with cancer present and scans without. And each and every one would need to be labeled so the model learns the distinction between cancer and cancer-free. You'd need to push extremely high quantities of labelled scan images through various training algorithms to create a model that could perform this diagnosis with an acceptable level of accuracy.

Have you got the right team?

You're getting into a major technical project here and that requires very specialist knowledge and experience. You cannot embark on building your own AI model without some key roles on the team.

Those crucial roles include:

- 1 Data Scientists for model designing, training and evaluation
- 2 Data Engineers for data collection, pipelines and storage management
- 3 ML Engineers for deploying AI models to production

You should also consider:

- MLOps Engineers for automating and monitoring AI workflows
- A specialist AI Product Manager to devise and manage the product strategy
- AI Ethicist - this is genuinely a role that is starting to emerge. They work to ensure fairness, bias reduction and compliance

Have you got the computational power?

The next consideration if you're starting down the path of building your own AI model, is whether you have the oomph to run it!

The computational power required for both training and deployment of your own AI model is considerable.

Although the demands will vary significantly depending on the complexity of the model, the size of the dataset, and the type of tasks the AI needs to perform, even the models at the smaller end of the spectrum will still need some hefty hardware behind them.

Let's start by looking at the processing power you need to actually train a model.

Training requirements

The training of a machine learning model requires immense computational power. We can't stress that enough.

This process involves running thousands or even millions of iterations of data through the model to adjust parameters and improve accuracy. You can't do that with your laptop.

The hardware needed typically includes:

- **GPUs (Graphics Processing Units):** High-performance GPUs, such as NVIDIA A100, H100, or AMD MI250, are essential for deep learning due to their ability to handle parallel computations efficiently.
- **TPUs (Tensor Processing Units):** Designed specifically for AI workloads, TPUs (offered by Google) provide even greater efficiency for tensor operations in large-scale deep learning tasks.
- **High-Core-Count CPUs:** For traditional machine learning models and preprocessing large datasets, CPUs with multiple cores (e.g., AMD EPYC or Intel Xeon) can be necessary.
- **Memory & Storage:** Large RAM (64GB+) and fast SSDs/NVMe drives are essential for handling big datasets without significant slowdowns.

For small-scale models, cloud services like AWS (SageMaker), Google Cloud (Vertex AI), or Azure Machine Learning can provide on-demand access to powerful GPUs and TPUs, so that might be something to explore if expensive hardware purchases are not an option.

Once your model is trained, you're still going to need a decent degree of power to deliver your AI experience through your product. Let's look at what's needed...

Hosting & inference requirements

Once trained, you need to deploy your AI model to serve predictions in real-time or batch processing scenarios. Compared to training, you typically need less computational power at this stage, but again, that will depend on the complexity and latency requirements of your application.

At a minimum, you're likely to need:

- **CPUs for Basic Inference:** Many AI models can run efficiently on high-performance CPUs for smaller workloads.
- **GPUs for Real-Time AI:** If the model needs to process high-throughput requests quickly (e.g., chatbots, real-time image recognition), deploying on a GPU-powered server ensures low-latency responses.
- **Edge Devices:** In cases where AI needs to run locally (e.g., IoT devices, mobile phones), specialized hardware like NVIDIA Jetson, Google Coral, or Apple's Neural Engine in iPhones can be used.
- **Cloud AI Services:** Cloud providers offer AI inference-optimized services such as AWS Inferentia, Google Cloud AI Platform, or Azure ML Inference for scalable hosting without dedicated infrastructure.

Have you got the technical infrastructure to support building your own AI model?

It's not just processing power that you'll need in place if you want to build your own model. There's the whole technical infrastructure that encompasses processing power, but lots more besides.

As well as processing power you need to have:

- **Large-scale data storage:** Your AI model will be relying on large volumes of high-quality data. That requires proper data storage and management systems. You need databases, data lakes, data pipelines and versioning tools.
- **Efficient monitoring and maintenance:** You'll need the right infrastructure in place to help you log and track system health and spot performance degradation over time. And potentially automated retraining pipelines that help you update the model training to improve any performance dips.

Are the costs feasible for the business?

Given everything we've said about computational power and data storage, cost is going to be a big factor when building your own AI model. Neither of those things come cheap!

So, before you decide on this path, you need to be certain that the business can sustain both the upfront costs involved in building and training the model, and the ongoing costs of hosting and running the AI in your application.

And remember, if your AI product is successful, those hosting costs will scale upwards. So you need to anticipate the scalability of your costs and know whether that is feasible for the business and makes the product cost-effective.

When it comes to cost, you need to balance what you buy in terms of power and storage, with the acceptable performance level you need to achieve.

Overprovisioning hardware can be expensive, while underpowered servers can lead to slow inference times and poor user experience. You'll need to find the right balance where the costs are acceptable to the business, while the performance is enough for the user. If you can't find that right balance, then you

probably need to reassess whether building your own model is the right option for your product.

If any of that has put the frighteners up you (and we wouldn't blame you - building your own AI model is a MAJOR technical undertaking), then it's time to think about the alternative - using an existing AI model.

Using an existing AI model

This is definitely the more widely used approach to creating and launching AI products into the market. And there are a number of reasons why that is, but in a nutshell those reasons are:

- It's cheaper
- It's faster
- It doesn't require vast quantities of data
- It means you can spend all your time developing your application rather than building the model

There are a decent number of open source LLMs or API-based models available for use now, so you even have options when you take this route. We'll cover the different models as well as how to choose between them a little later.

But first, let's make sure you're clear on exactly how this works...

How does it work when you use an existing AI model?

If you want to offer an AI product, or AI functionality, but building your own AI model does not make sense (as it rarely does), then you're going to need to lean on one of the many existing AI models and build your application on top of that.

There are two approaches to choose from:

- 1 Self-hosting an open source LLM (Large Language Model)
- 2 Using an API-based LLM

Self-hosting an open source LLM

With this route you will need to consider computational power as you're hosting the model yourself. Although you won't need the extreme power required to actually build and train a model, you are going to need some degree of processing power and data storage if you want to handle the hosting in-house.

How you host the model is up to you. It can be on-premise (using GPUs), cloud-based (through Amazon AWS, Google Cloud, Azure etc) or you can use edge deployment, running smaller models on devices like Jetson Nano or Coral TPU.

The advantages of self-hosting are a greater level of control over the model, with the ability to customize certain elements. This is also a better route if data privacy is likely to be a major concern for you and your users as you're keeping all the data on your own server.

Open source LLM options include:

- Meta's LLaMA
- Mistral AI
- Falcon
- Bloom
- GPT-J & GPT-NeoX

API-based LLM

The alternative to hosting a model, is to simply use the API of an existing model, hosted elsewhere, to make calls and provide your responses.

Here you have no heavy infrastructure requirements and you can typically get set up much faster.

However, you will be sacrificing control and customization of the model and you will need to think about the costs associated with each and every API call.

API-based LLM options included:

- OpenAI's GPT-4 / GPT-3.5
- Anthropic's Claude
- Google's Gemini
- Mistral AI's API

How do you know which AI model to use?

You should consider self-hosting if:

- Data privacy is paramount
- You have the existing infrastructure and processing power, or know you can finance it
- You need the highest level of customization of the model

You should consider an API-based 'managed service' if:

- Speed to market is paramount
- You can make your AI product or feature work through fine tuning rather than actual customization of the model
- You can support scalable per-request costs as your product usage increases

You can also adopt a hybrid approach, testing your concept and getting early market validation of your AI product through using an API-based model, then switching to self-hosted when you're confident your product has legs!

Also, you don't have to choose just one particular model and stick to it, especially if you're opting for the API-based route. You can switch between multiple providers, making API calls to different models in different circumstances. Whether that's based on which model is better at which task, or which model has the best performance at the time.

Considerations when using an existing AI model

OK, if this direction is sounding like the right fit for your product, what do you need to think about before making the final decision?

Let's run through the common considerations so you understand what you're getting into.

The costs

You have to pay for this! If you're going down the API-based route, someone else is footing the bill to power and host the model, plus all the work and data they put in to build and train the model. So they're going to charge you for usage. Obviously.

So, you need to think about how you handle those costs. How do you ensure this is scalable for the business? This normally comes down to how you price your AI product or features. Luckily there's a whole chapter in this ebook on that very

subject, so make sure you keep reading for more on how to make these costs work.

If you're leaning more towards the self-hosted option then you need to make sure you've weighed the costs of hosting up against the cost of the API-based option to make sure it's worth it.

Our CTO here at ProdPad, Simon Cast, crunched the numbers on this and believes you're looking at around \$13,000 of OpenAI API calls per month before it becomes more economical to host a model yourself.

Performance issues

You are completely beholden to the performance and stability of a third party model. It's as simple as that. There will likely be outages, downtime, and slow responses, and you'll need to have considered that in advance and think about how you'll handle that.

You can deal with the risk of sub-optimal performance through technical solutions, or customer care.

Technical solution

You could consider a process that routes the API call to a different AI model provider if one fails. This is probably worth looking at if you think a dip in performance would be unacceptable to your users or significantly impact your product's overall perception and performance.

Customer care solution

The alternative is to make sure you set the right expectations of the AI's performance across the touchpoints you have with users - from your marketing messaging, to your in-app copy and Customer Support communications.

It's also worth training your customer-facing teams so they know how to explain any AI performance issues with users, mitigating any major complaints or frustrations.

Dependency on a third party

If you use someone's AI model to power your product or features, you're introducing a dependency on a third party and that comes with risk to the business.

Since the performance of an external AI model will have such an impact on the performance of your product, you're removing a

significant amount of control from the business. This can make business leaders and board members nervous.

However, the chances are this wouldn't be the first time you're reliant on a third party. Your business is likely to have a number of different third party risk points - from hosting providers, to CRM tools. Use of third party providers is fairly standard practice these days, what's important is knowing how to spread that risk and have mitigation measures in place.

The model doesn't remember

This is an important point. When using existing AI models don't forget that the model won't remember! These AI models are 'stateless' - they have no memory built into the model itself. If memory needs to be captured it has to happen within the application you build.

So, if you need memory - for example if you need recall for a conversation bot - you'll need to build that into your product.

Knowledge cut-off

With most existing AI models, there will be a knowledge cut-off date that represents the end date on the data the model was trained on. In simple terms, the model won't have learnt anything new since that cut-off date.

You won't have the ability to patch a model with new or custom information. So if it's important that your AI tool is taking custom information into account, then you're going to have to layer on retrieval-augmented generation (RAG).

This involved capturing new knowledge (maybe from the prompt the user has given, or from some context data they've included) and placing it in a store. You'll then need to give the AI a way of drawing on the information in this store to inform their response.

Remember, when using an existing model you will never be able to put new information into the model, you can only fine tune it with system instructions, providing other content sources for them to draw on each and every time they're called upon to complete a task.

You need to fine tune

Do not underestimate the amount of work involved in fine tuning an existing AI model to ensure the outputs it creates in your product are relevant and accurate.

Although you can't train the model, you can overlay system instructions within your application that feed the model with guidance on how the output should be presented.

You need to factor in considerable time to feed the model with prompts and ideal response examples to help hone and optimize the outputs, ensuring your application of this AI model is relevant and valuable in the context of your product and your users.

A great example of this being done well is our very own ProdPad CoPilot. CoPilot is an AI tool specifically designed for Product Management and Product Teams, and, as such, the team here at ProdPad invested close to two year priming, fine tuning and prompting the AI model behind CoPilot, painstakingly feeding it system instructions based on real Product Team experiences. [Read more about the process that went into building CoPilot](#) and see what you can apply to your own AI product.

Which way will you go?

Now you know the core considerations of both approaches, which route is looking the most likely for you and your product?

It's best to have a good idea about your approach to the AI model that will power your AI product before we move on to the next chapter - Pricing AI. After all, the costs you incur to run your AI will have considerable bearing on your pricing decisions.



Pricing AI

How you price your AI product is a key decision when integrating AI into your offering, whether as a standalone solution or as part of an existing product. In the latter case, it's certainly not as simple as copying and pasting your current product pricing strategies. AI requires a fresh approach.

When launching an AI product or feature, you're going to need to work out your approach to **AI monetization**.

What is AI monetization?

AI monetization is the process of turning AI-powered features into revenue, whether as a standalone product, an add-on, or part of an existing offering. Some companies charge directly for AI functionality, making it a premium upgrade or pay-per-use feature, while others bundle AI into existing plans to drive adoption and retention.

There are two main ways to price your AI product:

- **Direct Monetization** – Charging explicitly for AI functionality.
- **Indirect Monetization** – Using AI to improve engagement and retention without charging for it separately.

Let's look at these two different ways to price your AI.

Direct AI monetization 💰

Direct AI monetization means explicitly charging users for AI-driven functionality. This approach makes sure that AI generates direct revenue, whether as an optional upgrade, a standalone product, or a core part of a pricing shift.

Here are the three main strategies within direct AI monetization:

- **AI as an Add-On:** Users pay extra for AI-powered features that enhance, but don't define the product. Best for products where AI adds value without being core to the offering.
- **Standalone AI Product:** AI is the product, with users subscribing or paying per use. Best for products where AI is the main value driver, not just an enhancement.
- **Bundled with a Price Increase:** AI features are included in existing plans, with prices adjusted accordingly. Best for products looking to boost value without separate AI upsells.

Indirect AI monetization

Indirect AI monetization focuses on using AI to improve user experience, engagement, and **user retention** rather than charging for it explicitly. Here you're utilizing AI as a way to make your product more compelling in order to drive growth.

Here are three common approaches to indirect AI monetization:

- **Bundled Without a Price Increase:** AI features are included in standard plans for free, driving acquisition and differentiation. Best for companies prioritizing long-term growth, loyalty, and competitive edge.
- **Freemium AI:** A basic AI version is free, with premium features behind a paywall, creating an upsell path. Best for companies showcasing AI's value upfront to convert engaged users.
- **Completely Free AI Features:** AI tools are offered at no cost to boost usage, activation, and loyalty. Best for platforms using AI to drive engagement and retention.

AI pricing models

Once you've decided whether to charge directly or indirectly for your AI product, the next step is determining how exactly your customers will pay for it. Your AI monetization strategy sets the foundation, and now it's time to translate that strategy into tangible pricing models.

While there are several options to consider, they generally fall into two distinct categories:

- 1 **Subscription-based AI pricing**
- 2 **Outcome-based pricing**

Subscription-based AI monetization

This is the most common model in SaaS, where AI features are included in a recurring pricing structure. This model works well for businesses looking for predictable revenue and a scalable growth strategy.

When using a subscription-based AI pricing model, you can choose:

- **Seat-based pricing:** Pricing is determined by the number of users accessing AI features. For example, charging \$50 per user/month for AI-powered automation tools.
- **Skill-based pricing:** Pricing varies based on the complexity or level of AI capabilities offered. For example, basic AI assistance is included in lower-tier plans, but advanced machine-learning capabilities require a premium upgrade.

Outcome-based AI monetization

Instead of charging upfront or per user, outcome-based pricing ties AI costs to measurable results. This aligns value with customer success but requires clear performance metrics.

When charging by outcome, this can come in the form of:

- **Usage-based pricing:** Customers pay based on how much they use AI features (e.g., API calls, queries, or data processed). This could be a chatbot platform charging \$0.01 per AI-generated response.
- **Output-based pricing:** Customers pay based on the volume of AI-generated outputs (e.g., reports, content, predictions). This might be a generative AI tool charging per 1,000 words generated, or blocking access once you've used a number of credits.
- **Outcome-based pricing:** Customers pay when AI delivers a tangible business result, like increased revenue or cost savings. This one is specifically suited to B2B businesses and can be like an AI-powered hiring tool charging per successful hire.

How to choose the right AI pricing strategy

When choosing how to price your AI, you need to decide between **direct** or **indirect monetization**, and then **subscription** or **outcome-based pricing**. Here's the ideal use case for each overall strategy and model:

- 1 **Direct monetization** is ideal if your AI offers high-impact, standalone value. This is best when AI delivers measurable ROI and you need to cover high operational costs. However, it might limit adoption if users expect bundled pricing.
- 2 **Indirect monetization** works when AI enhances the core product, driving engagement without charging extra. It's great for improving functionality (think smarter recommendations) but may not fully capture AI's value.
- 3 **Subscription pricing** works well for AI that's deeply integrated into the product, providing predictable revenue. You can choose between seat-based pricing (based on the number of users) or skill-based pricing (based on AI complexity).
- 4 **Outcome-based pricing** is best when AI delivers clear, measurable business outcomes, like cost savings or revenue generation. This can be usage-based or output-based, aligning price with value delivered.



How to manage an AI product

Let's go through all the things you would have done before building an AI product:

- Been through all the considerations of building an AI product? Check.
- Understanding the unique factors you'll face based on whether you're building a new AI-native tool or adding AI to your existing product? Yup.
- Figured out if you're building an AI model or using an existing one? Affirmative.
- Chosen your AI pricing strategy? Of course.

Awesome, well after all that you're ready to roll out your product and introduce it to the masses. Nice work. But wait, your job isn't over. Building an AI product is one thing, it's a whole different story managing it throughout its entire lifespan.

Now that you have an AI product, you're suddenly an AI Product Manager. That means there's a few more responsibilities that get added to your plate. Here's a look at how to properly manage your AI product once you've got it off the ground.

Make sure you understand the training data of your model

Your AI tool is only as good as the data it's trained on. When managing an AI product, it's a core responsibility to continue to monitor the source of this data and assess any potential biases and how it affects your tool's ongoing performance.

Regularly assess your dataset to see if it needs updating or cleaning to improve the results from your AI.

Be clear on the risks of your AI tool and how to mitigate them

Make sure you're always aware of the risks when offering an AI product. Stay abreast of misinformation or compliance issues, and create strategies to help mitigate these risks - be it through human oversight, improved training data, or clearer guardrails for the model.

We go into further detail on the challenges of AI and how to fix them in our final chapter.

Keep your AI literacy up to date

AI keeps changing. You may be an expert one day, and then a dummy the next if you don't stay up-to-date with industry changes and new advancements. As an AI Product Manager, you need to stay informed about changing capabilities and best practices.

Manage continuous learning and adaption

AI models aren't static - they degrade over time as the world changes. You'll need to schedule periodic retraining, monitor data drift, and ensure your model keeps up with user needs. Keep an eye on key performance indicators (KPIs) to track whether updates are improving or worsening results.

Keep constraining the outputs

AI can be unpredictable, and left unchecked, it might generate misleading or inappropriate content. You need to 'constrain' your AI tool and continuously refine your system instructions to improve accuracy and usefulness.

Use techniques like prompt engineering, reinforcement learning, and strict validation rules to prevent hallucinations and ensure AI-generated outputs align with business goals.

Monitor usage

AI isn't cheap. Keep track of how users interact with your AI functionality and analyze trends in performance and adoption. Cloud-based AI models can be costly, so monitor API calls, processing time, and storage use to prevent runaway expenses.

Make sure your customer facing team can answer questions

AI is complex, and customers will have questions. You don't want your Customer Success team to make your business look a fool when stumped with a security question.

Your Customer Success and Sales teams need to understand how the AI works, its limitations, and how to communicate this effectively to users. Provide ongoing training and FAQs so they can confidently handle AI-related inquiries.

Skills needed to be a good AI Product Manager

When managing an AI product, you're going to need the right skillset. As we've explored, AI products demand different things from you as a PM. Here's a look at the skills you need to develop:

- **Strong Product Management foundation:** Like any PM, you'll manage the full product lifecycle, work with cross-functional teams, prioritize features, and align stakeholders. You still need that solid foundation as an AI Product Manager.
- **Understanding AI & machine learning:** You don't need to build models, but you should understand how they work and be able to translate between technical and non-technical teams.
- **Prompt engineering:** AI models rely on well-crafted inputs. Knowing how to design prompts and evaluate outputs is essential for fine-tuning AI features.
- **Data-driven decision making:** AI thrives on data. You'll need to analyze insights, collaborate with data teams, and make informed product decisions based on experiments and trends.
- **User-centered design:** AI should enhance user experiences, not complicate them. Understanding how to integrate AI seamlessly into intuitive workflows is key.
- **Problem-solving & innovation:** AI unlocks new possibilities, but it takes creative thinking to apply it effectively. Iterate, experiment, and refine as user needs evolve.
- **Ethical considerations & responsible AI:** AI introduces challenges like bias, fairness, and privacy. Ensuring transparency and responsible AI development is a critical part of the role.

Not currently managing an AI product but really want a slice of the action? You can find out more about how to become an AI Product Manager in our guide.

AI Product Manager: Everything You
Need to Know To Become One

[Read now](#)



So those are the major things you need to focus on when approaching the ongoing management of your AI product or features. We mentioned there how it's important for an AI Product Manager to understand all the risks and challenges of using AI in your product... so let's delve deeper into that.



Challenges of AI for Product Managers

Managing an AI product won't be easy. This type of technology brings up a hell of a lot of challenges that you're going to have to navigate. Here's a look at the ones that are likely to crop up as you incorporate AI into your product.

AI is probabilistic, not deterministic - get comfortable with uncertainty

Most software is deterministic, it gives the same output every time. AI? Not so much. It's probabilistic, meaning responses can vary, even with the same input. That's fine for some products but risky for others.

Before rolling out AI, ask yourself: What happens when it's wrong? Is it just a mild annoyance, or does it create legal, ethical, or even life-threatening risks?

The fix? Set clear expectations with users. Use disclaimers, confidence scores, or explainability features that detail the AI model's thinking process to improve trust and accountability. If the risk of errors is too high, consider limiting AI's role in high-stakes decisions.

AI will make mistakes - plan for the worst-case scenario

No AI is perfect, so what happens when it gets something wrong? Do users have a way to fix or report errors? If a bad AI response means a broken experience, you'll need a safety net (think: manual overrides, clear disclaimers, or feedback loops).

The fix? Build in fallback mechanisms. Give users easy ways to correct AI mistakes, provide human support when needed, and log errors to improve the model over time.

Bias lurks in every dataset - know your AI's training data

AI models inherit biases from their data. If you don't check for them, you could end up reinforcing stereotypes or make unfair decisions. Make sure you understand your AI's limitations and have a plan to audit, tweak, and refine results over time.

The fix? Regularly audit your AI outputs, test for bias, and diversify training data. Be transparent about AI limitations, and give users the option to provide feedback on unfair or inaccurate results.

Set user expectations - AI isn't magic, it's can blank sometimes

AI takes time to process, and nobody likes staring at a blank screen. Use smart UI design - loading indicators, progress bars, or friendly messaging - to keep users in the loop while they wait. A little transparency goes a long way.

The fix? Improve perceived speed with skeleton screens, animations, or estimated wait times. If long delays are unavoidable, consider background processing or asynchronous updates to keep users engaged.

Balance accuracy, speed, cost, and experience - pick your trade-offs wisely

Want AI to be lightning-fast? That might mean sacrificing accuracy. Need it to be spot-on? It could be expensive and slow. There's no perfect balance, just the one that makes sense for your product. Figure out what matters most to your users and optimize accordingly.

The fix? Test different configurations to find the sweet spot. Optimize inference models, use caching for common queries, and adjust processing based on user needs (e.g., high accuracy for critical tasks, faster responses for casual use).

The risk, implications, and legality of data usage - tread carefully

AI needs data to function, but how that data is collected, stored, and used comes with serious risks. Users need to trust that their data is safe, but questions about provenance, copyright, and biases in training data make this tricky.

Then there's the legal gray area - if AI makes a bad recommendation, who's responsible? Plus, there's always the risk of bad actors misusing AI, which could put your company in hot water.

The fix? Be transparent about data usage. Clearly communicate how data is collected, stored, and protected. Work with Legal Teams to navigate evolving AI regulations, and put safeguards in place to prevent misuse (like content filters, abuse detection, or ethical AI guidelines). If your AI makes recommendations, make sure accountability is built into the system. Users should know where suggestions come from and have the final say.

Conclusion

As we've explored throughout this ebook, integrating AI into your product (or building a standalone AI tool) is no small feat. Whether you're enhancing an existing product with AI capabilities, launching something entirely new, or developing a custom AI model, the journey demands significant resources, strategic planning, and ongoing management.

But if AI aligns with your product vision and genuinely addresses the problems you're solving, it can be transformative. The key isn't just adopting AI for the sake of it: it's about using it to create real, tangible value. Your AI strategy should be deeply connected to your product goals, improving your value proposition and expanding what's possible for your users.

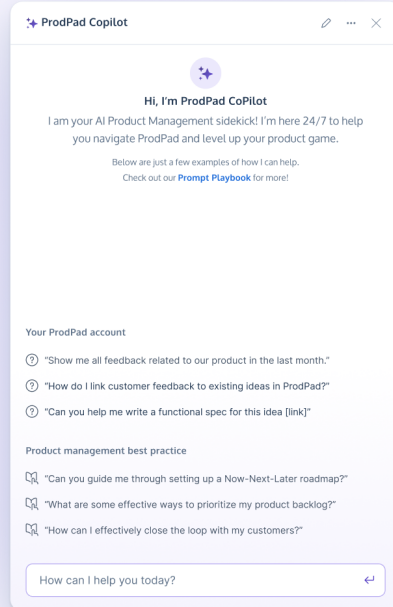
AI is no longer a futuristic add-on; it's becoming an expectation. When used correctly, it can give your product a competitive edge. However, success requires more than just jumping on the AI bandwagon. You need a long-term vision, thoughtful execution, and a commitment to continuous learning and improvement, exactly like any other major feature update or release.

The technology will keep evolving, and so must your product. By refining your AI models, optimizing performance, and maintaining transparency with your users, you can build not just an AI-powered product, but a future-proof one.

Yes, AI is challenging, but when done right, it's a game-changer. Embrace the challenge, and you'll position your business for long-term success in an AI-driven world.

Work better and faster with CoPilot in your corner

The world's best Product Management AI tool



Try for free

Ask me anything

I know your product and work so you or your stakeholders can ask away!

Do we already have an idea about a browser extension?

Why did we decide to pause the Live Chat idea?

What are the biggest themes from our customer feedback this month?

Give me your tasks

Let me generate your documentation, summarize your Feedback and more.

Add Idea

Idea title

Give users the ability to integrate their calendars

Idea description

to integrate their calendars

Generate description

I've generated a description for your idea. What would you like me to do next?

Refine with AI

Write User Stories

Add functional specs

Follow up...

Let me help you make the smartest decisions

From kick-starting your ideation, to surfacing your Feedback themes. I can make informed suggestions that have you solving problems, hitting your goals and driving better results.

Now

Generate

Next

Later

Increase engagement

Can we improve effectiveness by providing integrations with external services?

Grow our userbase

Can we make it more likely for people to choose our app if it works alongside other commonly used apps?

Increase

Can we ap are able to abilities?