

# Product Discovery: A Practical Guide for Product Teams



In this hands-on guide, I will show you how to go beyond cliché advice, like “just talk to users more often,” to make sure the practice of Product Discovery helps teams to make real progress.

These are the exact techniques that I use to help Product Teams around the globe navigate the uncertain problem space of their user segments and identify solutions that are worth building through evidence-informed prioritization and testing.

In this in-depth guide, you'll learn:

- What Product Discovery is and how it connects to Product Management domains like Product Strategy, Scrum, and OKRs
- How to structure your Product Discovery without losing the ability to explore and course-correct
- How to navigate and combine the key elements of Product Discovery, without sticking to rigid formulas and templates
- How to implement effective, yet lightweight Product Discovery techniques around alignment, research, and testing
- Lots of advanced tips, strategies, and tactics.

If you want to make sure that you're building the right product for the right audience, you'll love this guide.

Let's get started.



# Introduction

When crafting roadmaps for upcoming cycles, you should be able to make the case for the themes you and your team want to work on based on evidence and strategic fit, not just gut feeling. Most often, those topics result from epics you've worked on before, your Product Strategy, incoming user feedback, or the Objectives and Key Results (OKRs) for your team.

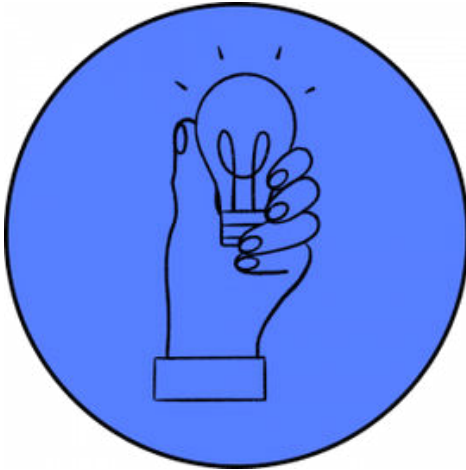
By then, it's pretty clear what to work on from a high-level perspective. But as a Product Manager, you might wonder exactly which problem needs to be solved next, and for whom? How can you focus on ideas that provide the most impact? This is when you need a well-prepared and insightful Product Discovery process.

The goal of this guide is to show you how expansive Product Discovery can be while setting you up with specific strategies and tactics to execute your own Product Discovery process. The most effective way to start the implementation of your own approach is to focus on a set of techniques and habits you want to internalize and that work for your specific environment.

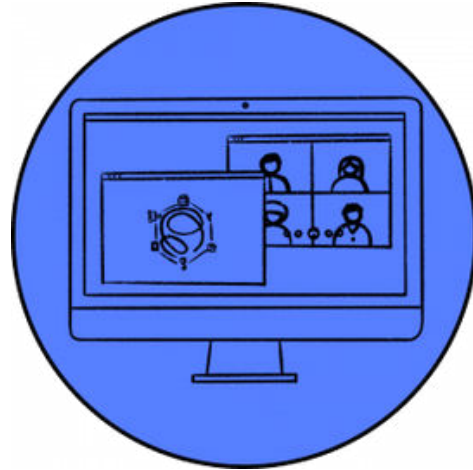
Don't treat this guide as a step-by-step blueprint for every Discovery. View it as an inspiration to structure your thinking and choose what matches your unique situation. It is designed to supplement my Adaptable Product Discovery Course, which helps you design and implement your own Product Discovery process.



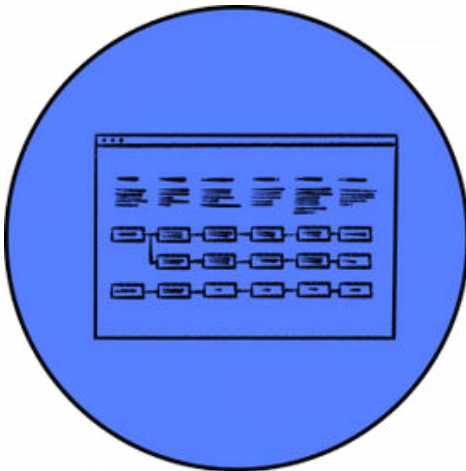
# Content Overview



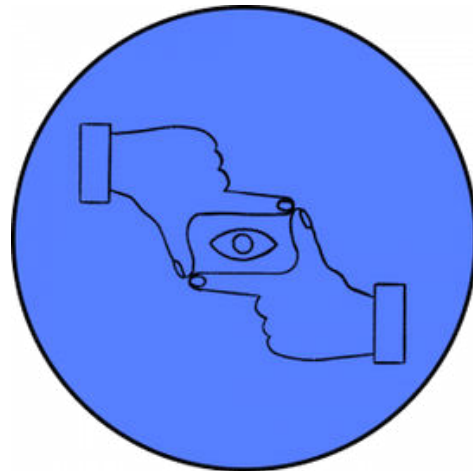
Product Discovery  
Fundamentals to get Started



Product Discovery Phases  
and Team Participation

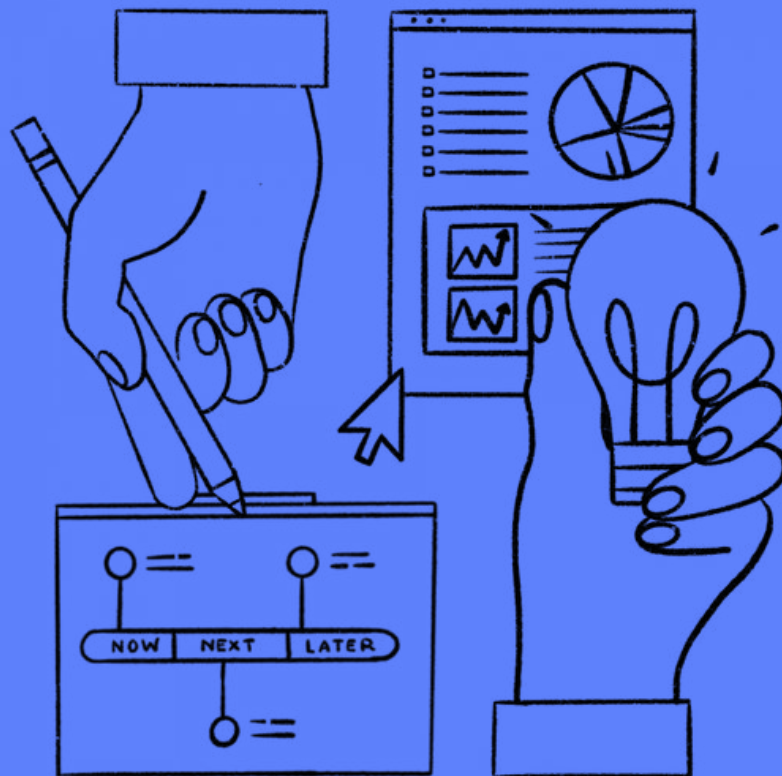


Product Discovery  
Techniques and Frameworks



Discovery Practices  
Reality Check

# Product Discovery Fundamentals



In this chapter, I want to introduce the basic concept of Product Discovery and highlight how Product Discovery connects to some of its adjacent processes.

Your Main Takeaways From This Chapter:

- Understand what Product Discovery is
- How Product Discovery relates to Scrum to form Dual-Track Agile
- Connect Product Discovery to Product Strategy, OKRs, and Product Roadmaps



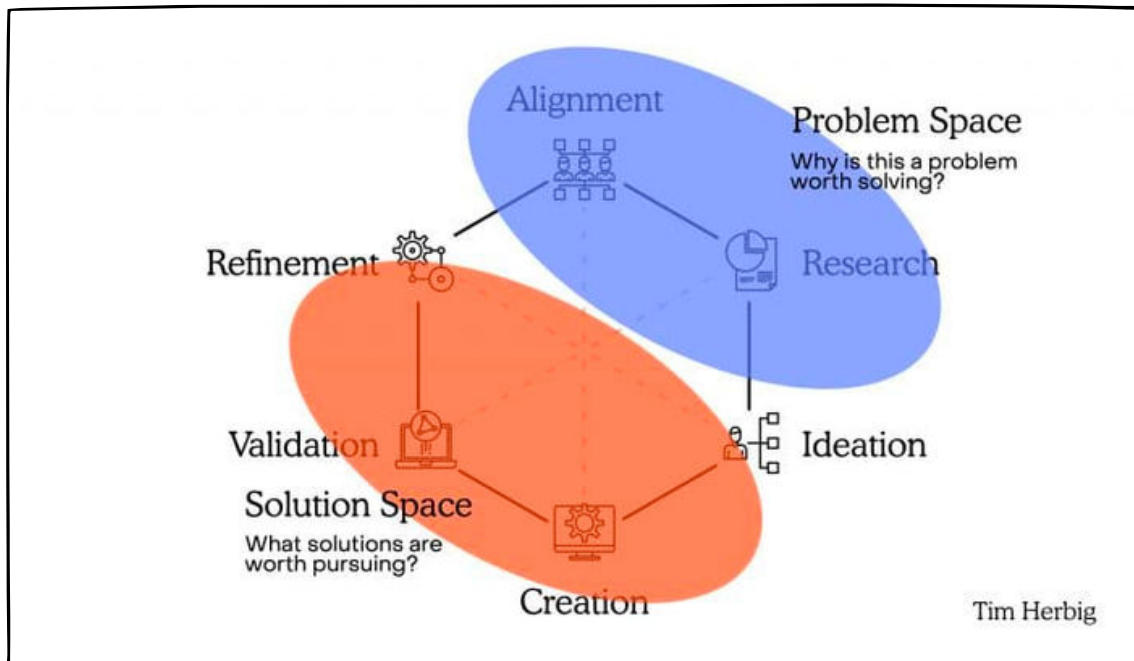
# What Is Product Discovery and Why Is It So Important?

At its core, Product Discovery is the evidence-informed process of reducing uncertainty as you find problems worth solving and solutions worth building. It emerges through a series of nonlinear activities, conducted as a cross-functional team.

At its core, Product Discovery is the evidence-informed process of reducing uncertainty as you find problems worth solving and solutions worth building. It emerges through a series of nonlinear activities, conducted as a cross-functional team.

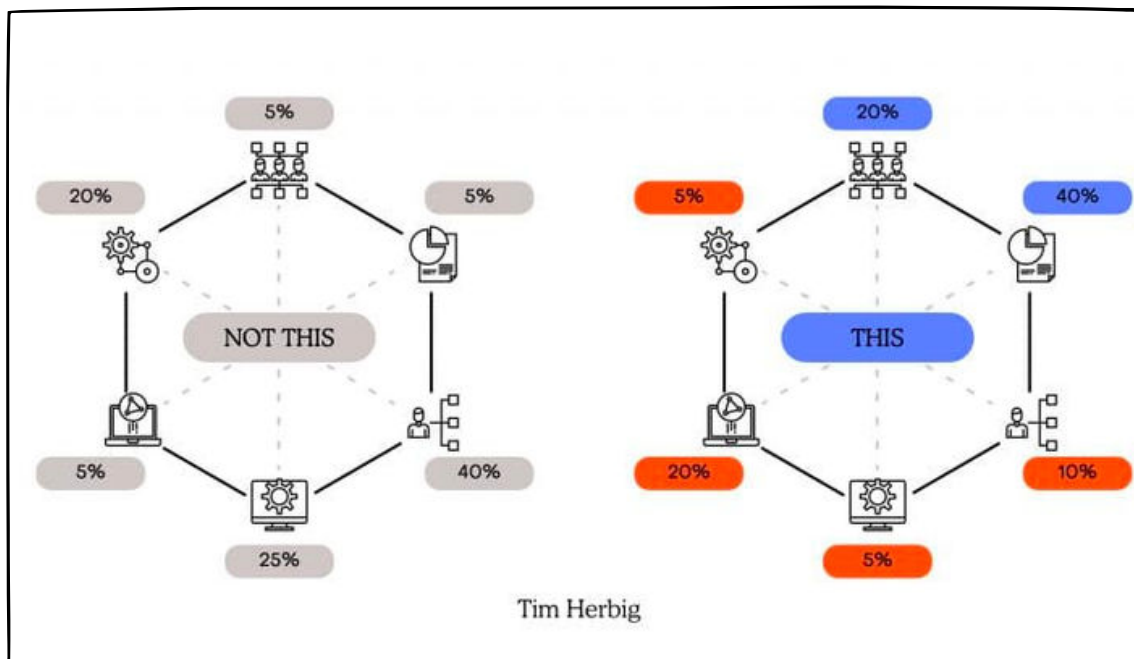
Product Discovery typically describes a (flexible) period during which you and your team focus on building the right thing as opposed to building the thing right, which happens during Product Delivery. Product Discovery can have many forms and structures, as we'll discuss later.

Typically, Product Teams focus either on the problem space or the solution space. In other words, they are either still trying to understand whether a problem exists for their users, customers, or stakeholders or they are focused on executing a matching solution. Though working in both spaces is necessary to satisfy users and support the business, teams often get the allocation of time and attention wrong. Product Discovery is, or at least should be, mostly about the problem space.



Differentiating the Problem Space and Solution Space of Product Discovery

So, though it might be tempting to jump straight into discussing and designing shiny solution ideas, it's important to stay disciplined about what to work on. Even though I'm a big supporter of individual Product Discovery paths, I believe that outlining and aligning on the problem space is a crucial activity, which you should rarely skip.



Reflect on and be mindful of how you allocate your time and efforts between the problem space and solution space of Product Discovery.



## Avoiding Alibi-Discovery and Owning the Problem Space

For those of you who are not familiar with it, Alibi-Discovery describes the process of developing an existing (top-down) idea that is already set to be next in line for the feature factory. Here's the quickest way to recognize if a team is doing Alibi-Discovery: They start their Discovery mandate with ideation sessions.

If you use the definition of Product Discovery above, it should become clear why ideation shouldn't dominate the Discovery process. Instead, Product Discovery should turn generic, company-level, Impacts into tangible Outcomes for Product Teams to focus on.

Here are some of my favorite probing questions to see how well a team understands its problem space:

- What are the three pieces of first-hand evidence that support the notion that this problem is worth solving?
- When was the last time you had valid interactions with users from your target audience?
- Can you name the specific changes in behavior your user would have to adopt so that they can achieve their goals and make progress?
- Do you have quantitative AND qualitative insights into the problem?

Product Discovery is not necessarily about the number of interviewed users or how many experiments you ran. Identifying the right problem space to explore and genuinely understanding that problem can happen in many shapes in forms. If you only switch the number of shipped features for the number of user interviews your team will just end up on a different hamster wheel, instead of focusing on the right uncertainty-reducing activities at a given point in time.



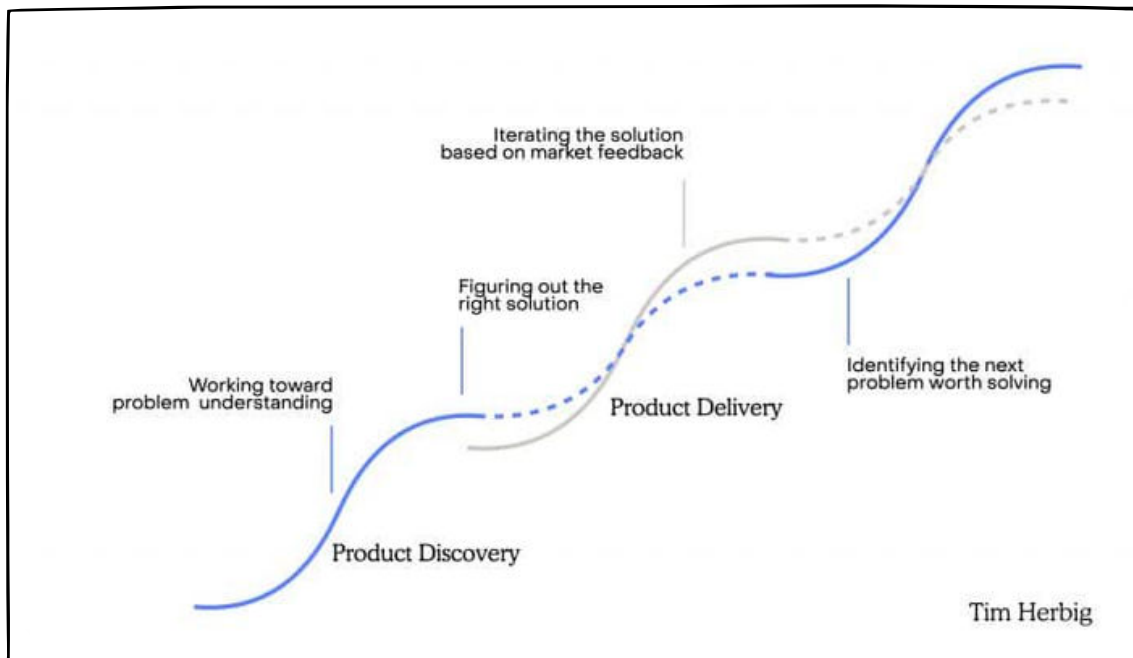


# Connecting Product Discovery to Scrum and Product Delivery

In general, Product Teams are always in some Product Discovery mode. This is because if you're set up as a truly customer-centric organization, you want to expose as many team members as possible to user feedback as often as possible. That will ultimately lead continuously to new insights in the user's needs.

Most Product Managers face a scenario called Dual-Track Agile. This means that you need to simultaneously apply Agile principles (like cross-functional collaboration among Product, Design, and Engineering; user-centered thinking; and iterative improvements) to the Discovery and Delivery tracks of your product.

Delivery and Discovery activities need to overlap and play together. Both are continuous activities, but their peak efforts behave more like S curves.



Dual-Track Agile should be seen as alternating S curves with continuous highs and lows, instead of a rigid and monotonous way of working.





The results from both activity tracks heavily influence each other and are, to a certain extent, happening in parallel all the time. However, I believe it's important to acknowledge that both activities require a different dedication of time from the Product Manager, UX Designer, and Development Team. Though a Product Team should ideally start working together on a new Product Discovery mission, some team members will be more occupied with it than others. It's ok to temporarily focus more on Delivery or Discovery at a given point in time. But make sure that a Product Team has the skills and freedom to work on both aspects on their own time.

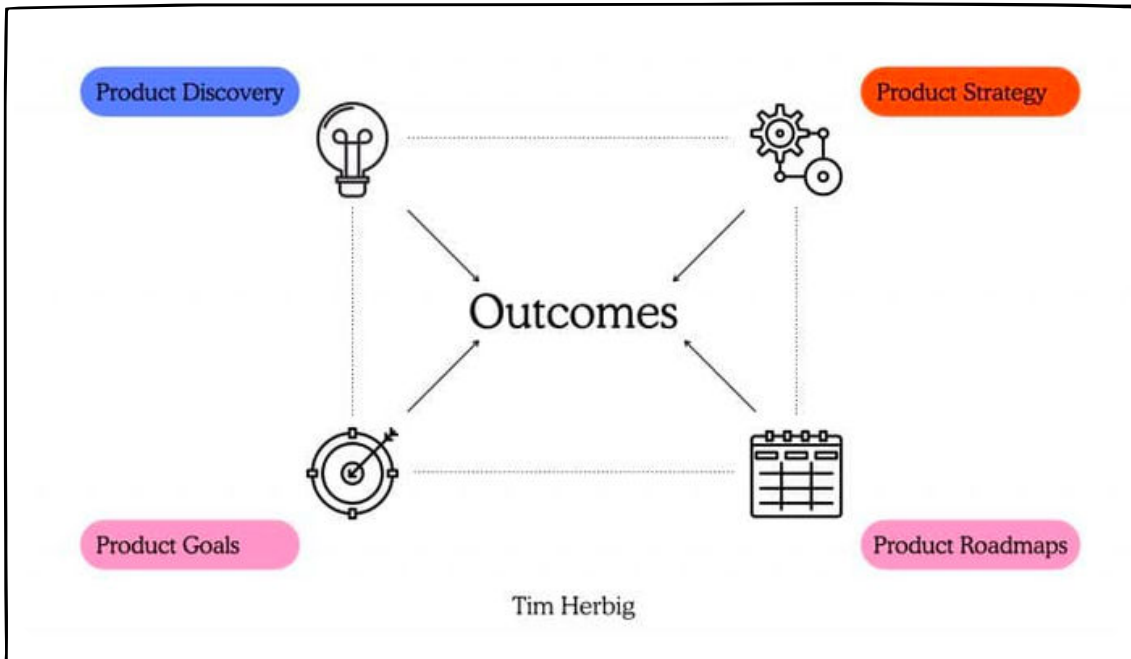
The inconvenient truth is that you probably work in a mix of Discovery and Delivery modes for most of the time with just the scale tipping more to one or the other direction at certain points.

The more uncertainty you have about which solutions your business goals need to bring to the table, the more attention you need to devote to Product Discovery. In general, I recommend changing your mindset from treating Product Discovery as a separate and clearly defined task. Instead, Product Discovery is a combination of activities to navigate the problem space and the solution space. And this combination needs to be adaptable and nonlinear. Product Teams need to adopt it as a habit.



# Adjacent Product Discovery Processes

Your ability to focus on Outcomes through Discovery doesn't depend just on your ability to talk to users or run experiments. Instead, it requires the joint effect of something I like to call the Outcome Quadrant. The Outcome Quadrant consists of the four key pillars that influence your overall focus on Outcomes.



Focusing on Outcomes in practice requires changes to and input from multiple Product Management domains that are part of the Outcome Quadrant.

Though Product Discovery is one of the key pillars, you also need an outcome-oriented approach to Product Strategy, Product Goals, and Product Roadmaps.

As a Product Manager, it is important that you understand what you can and should change and where to demand clarity. Product Discovery, Strategy, Goals, and Roadmaps are within your realm of responsibility. If your Discovery efforts feel slow or misguided, chances are it's not just because you need better interviewing skills or more a/b testing. Ensure these three adjacent processes are supporting your efforts as well.

Obviously, there are many ways to approach each of these three pillars.



## How Product Strategy informs Product Discovery Priorities

At its very core, Product Strategy is about expressing how you plan to compete in and win on the playing field you have chosen. It outlines elements like your strategic audiences, the alternatives to your product, and how to differentiate from it, as well as your Product Vision, and your company's capabilities and overarching goals.

Product Strategy is about expressing how you plan to compete in and win on the playing field you have chosen.

Product Discovery starts with understanding the strategic context and the Product Strategy patterns that matter. The point is not to create a McKinsey-level company strategy, but something to refer to when you have to make trade-off decisions as you navigate through the problem space of a Discovery mission.

This, of course, requires a clear understanding of what a Product Strategy is about compared to, for example, a Product Vision. Check-out this video, where I'll clearly differentiate Product Vision from Product Strategy and show how they relate to and benefit each other.

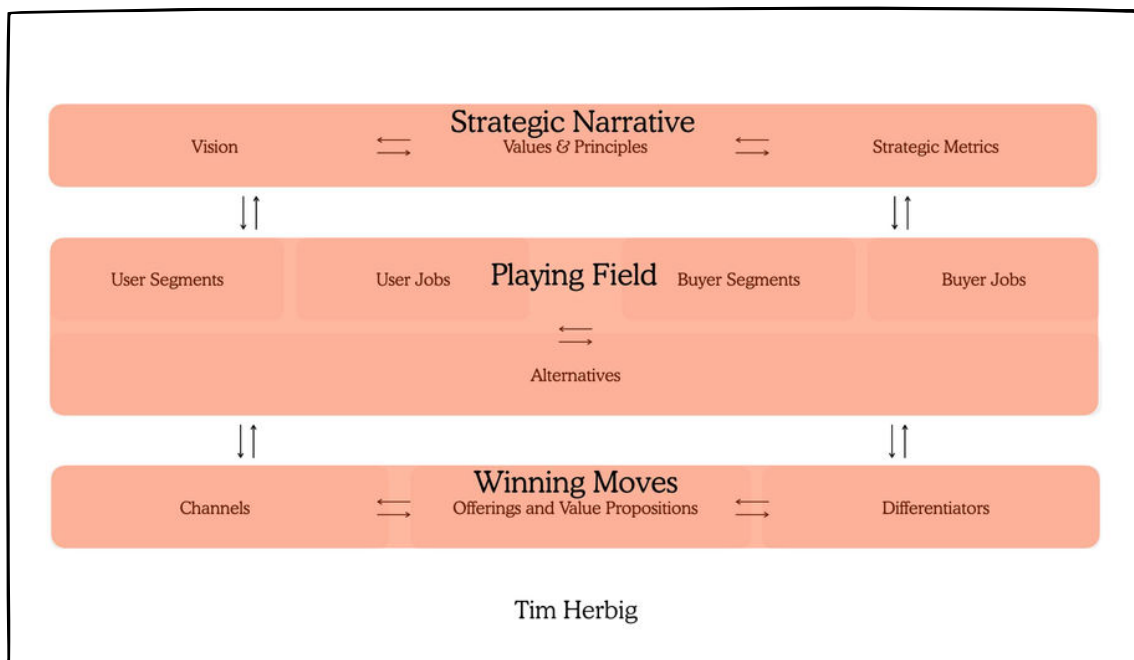




Without going into too much detail about Product Strategy, I want to outline the cornerstone questions it should be built on:

1. The Strategic Narrative tells you why you are targeting a specific Playing Field. What future are you trying to create and how is experienced value measured? Vision, Principles, and mid-to-long-term goals provide this context.
2. The Playing Field describes that market itself, including your possible choices and your competition. It is composed of particular user, stakeholder, or buyer segments, as well as the relevant jobs you want to help them pursue in the light of your Strategic Narrative. In addition, your playing field consists of alternatives that users and buyers are comparing you against.
3. The Winning Moves outline the choices you need to make to position your product to best advantage. To win on the Playing Field and avoid, like Roger Martin says, "playing to play," you have to articulate what offerings and value propositions will address the jobs and through which channels you can deliver them. In addition, how do you differentiate between alternatives?

Filling in Product Strategy Templates won't help you here: they encourage filling in the same type of information, in the same order and way of working. Looking for common Product Strategy patterns encourages a more flexible, iterative, and non-linear process.



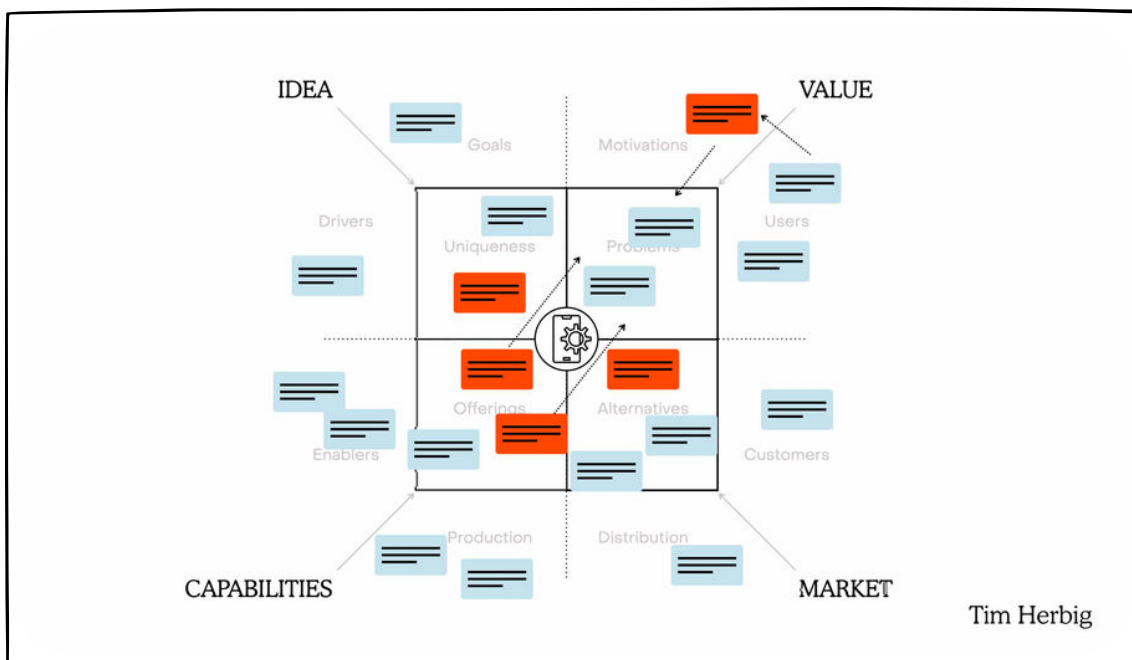
The ingredients of Product Strategy Patterns informing Product Discovery priorities



The gaps in your Product Strategy will drive the specific priorities of your Product Discovery.

1. Are you trying to close a performance gap within an existing playing field? This may require product iteration.
2. Are you trying to crack an opportunity gap? This may require a new market or a new vision.

Depending on the gap you're trying to close, your Product Discovery priorities will look different. For example, are you trying to understand (and ultimately serve) new jobs for your existing audience? If so, use qualitative and quantitative insights about existing or churned users and pick research tactics that fill the gaps, as opposed to exploratory interviews with entirely new audiences. Based on your company structure, revisit your Product Strategy stack for guidance on the gap with the highest priority for your product team.



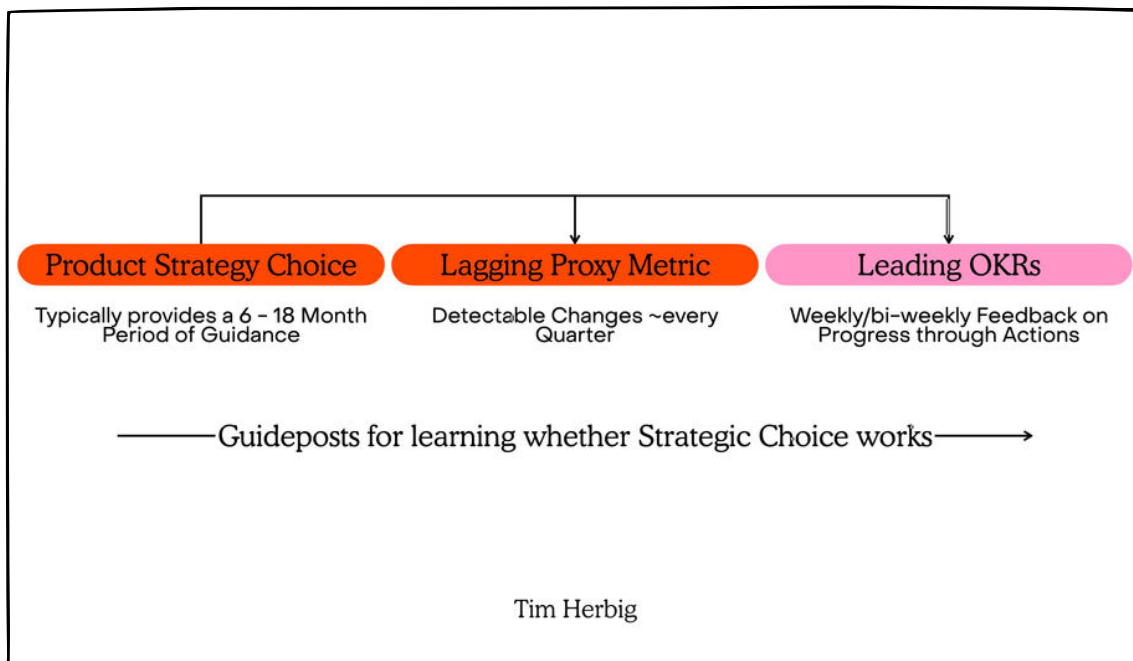
Visualizing the gaps in the various constellations of your Product Strategy using the Product Field canvas

You can learn more about how to shape and articulate your Product Strategy and connect it to the implementation of OKRs and Product Discovery in my Path to Product Strategy Course.



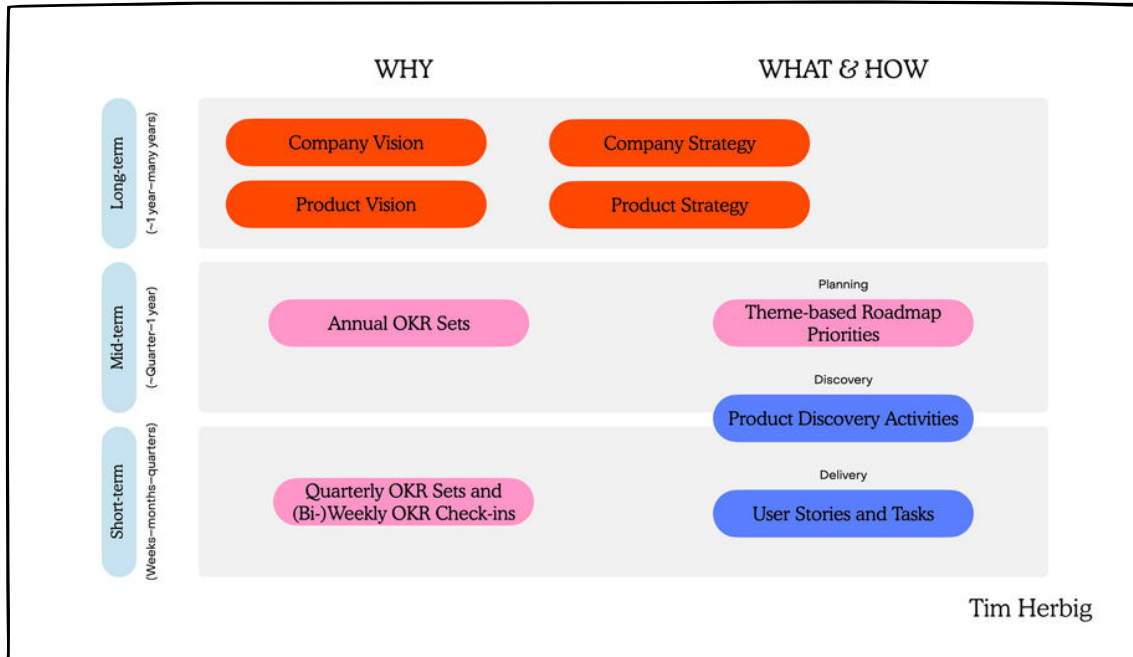
## How Product Discovery Relates to OKRs

Goals like Objectives & Key Results (OKRs) take the specific choices of Product Strategy and quantify what success would look like, so product teams can prioritize their day-to-day activities. Ideally, OKRs express a holistic perspective of how exactly success would look, in the form of quantifiable metrics.



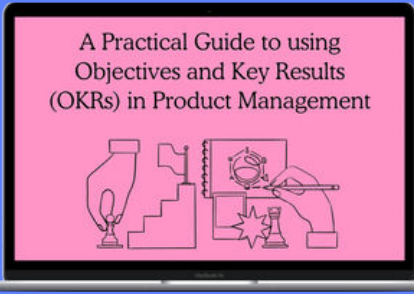
Connecting Product Strategy Choices to leading OKRs for Product Teams to structure Product Discovery and Delivery progress

Product Goals are often conflated with strategy (whether it's in B2B or B2C). But Strategy is the plan to achieve your goal—not the goal itself. One of the biggest challenges of using OKRs in Product Management is aligning the Discovery and Delivery cadence with goal cycles.



Connecting OKRs to Product Management Domains and Product Discovery Time Horizons

So, instead of trying to rush through Discovery, Delivery, and iterations within the same cycle, I believe that Product Teams need to lay the groundwork for upcoming Delivery-oriented OKRs and activities by anticipating Discovery-oriented, exploratory OKRs in a previous cycle. That is, as long as these OKRs are connected to an actual benefit you want—and don't get misused as a task list to keep the Product Team busy.



[Learn More](#)

## Deep Dive: OKRs in Product Management

If you're asking yourself how you can connect Product Strategy and your Product Discovery/Delivery efforts through outcome-oriented goals, this guide has the answers.





Making OKRs a truly useful practice for Product Teams that goes beyond pretty, but generic KPIs and task-lists requires going beyond the common "by-the-book" advice. My [Outcome OKRs for Product Teams Course](#) helps you to arrive at OKR systems that work for you and tie in with your existing product management processes.

## How Product Discovery Connects to Product Roadmaps

An often dismissed and underestimated artifact for getting inspiration and guidance about upcoming Discovery priorities is the Product Roadmap. Product Roadmaps can be a helpful format to make the priorities and activities you have chosen to pursue more tangible.

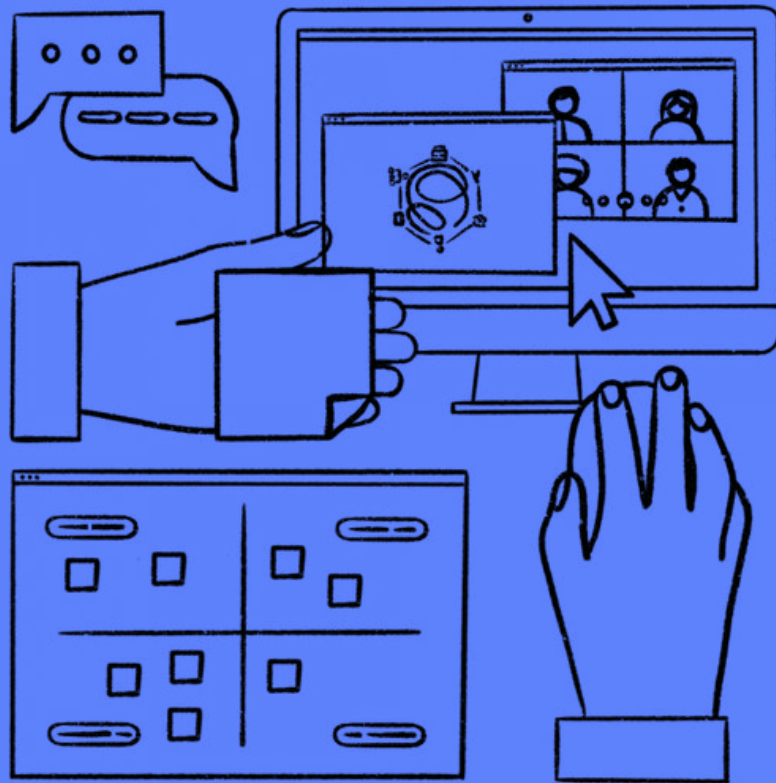
The way you handle roadmaps and plan “the work” directly influences the freedom and autonomy your team has for Product Discovery. When you have the resources to understand the problem space, but your roadmap says you need to deliver the new start page feed on March 31, you’re not doing Product Discovery. Instead, you’re polishing an existing idea.

When you have the resources to understand the problem space, but your roadmap says you need to deliver the new start page feed on March 31, you’re not doing Product Discovery. You’re polishing an existing idea.

When looking ahead for, let’s say, one year, Product Teams should have high confidence in the current quarter’s areas of focus, but should assume that they will learn a lot during the current quarter and that plans for subsequent quarters may need to change. That’s why a theme-based roadmap approach with loosely-defined time horizons, that become less granular the more you look into the future, is more fitting.

A roadmap like this should then be updated on a rolling cycle-based cadence and aligned to the activities of Strategy and Goal iterations.

# Product Discovery Phases and Team Participation



In this chapter, I will outline some of the most important phases teams should cycle through during Product Discovery, as well as discuss how to structure these activities.

Your Main Takeaways From This Chapter:

- Prioritizing and structuring your Product Discovery
- Who should participate at what stage of Product Discovery
- What are the key elements of Product Discovery.



Though every Product Discovery starts at a different stage, I find the model below helpful. The segmentation into certain phases also makes sense for structuring your thoughts and actions during continuous Discovery efforts.



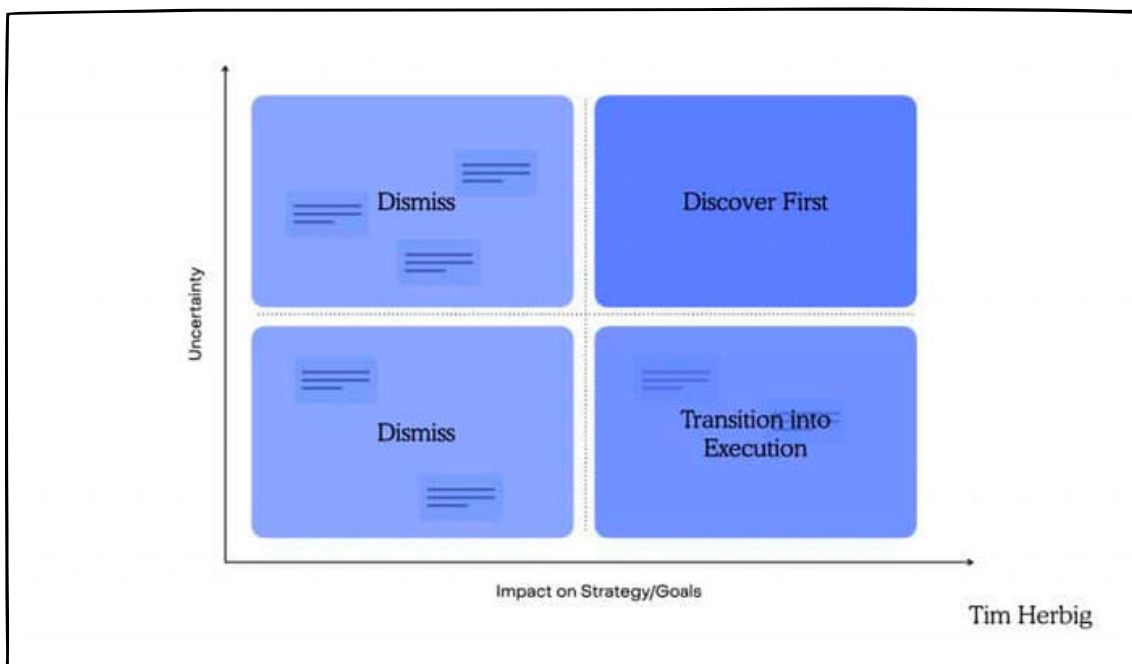
It's important to understand that you won't just walk through these phases one after another toward a pot of gold at the end of the rainbow. You might get stuck during one stage and need to take one or multiple steps back to pursue a different path—and that's ok.

Product Discovery is rarely a linear process. I've developed an [Adaptable Product Discovery Approach](#) that will give you the tools you need when you need them—and show you how to make them work for you and your company.



# How to Prioritize Product Discovery Opportunities

Product Discovery is about reducing uncertainty and increasing the confidence to invest resources in building a specific product—based on strong evidence, collected first-hand by product teams. This is why frameworks like simple Effort-Impact-Scoring fall short—because you don't know what kind of solution you should rate. I prefer a different grading:



Product Discovery Priorities should be defined along the lines of Impact and Uncertainty

By evaluating the uncertainty, you get a much clearer picture about which ideas you need to further explore during Product Discovery. The area on the upper right includes your next targets to evaluate. Everything on the lower right should be looked at from a Delivery-capacity perspective.

Prioritizing opportunities may seem rough to you—and that's ok. This is not about making a 100 percent accurate prediction; it's about a shared understanding within your team and company about which ideas to pursue.



You will gather more insights and evidence on what works and what doesn't as you move forward. Getting started is more important than being right since only the insights you collect through Discovery activities will yield results.

## How to Set Up Your Product Discovery Process

When it comes to planning and structuring an upcoming Product Discovery, you don't want to start with the day-to-day tactics. Instead, it makes sense to gradually zoom in.



As mentioned earlier, the majority of Product Discovery should be spent trying to understand the problem space. Though it might be tempting to jump straight into ideation sessions discussing specific solutions, it's important to take your time to clarify the problem you're trying to solve.

Though I don't believe in setting an artificial deadline for "completing" Product Discovery for the sake of it, I've seen the power of defining a timebox to set up the team for a "best effort" mission.



Process and structure are not the enemies of Product Discovery—as long as they are seen as options to choose from, depending on the context and phase of Discovery.

In my experience, a period of six weeks has proven to work quite well. Why six weeks?

- It doesn't try to predict how “everything” will play out (limited risk investment).
- It's easily seen as a starting point, rather than a final timeline.
- It leaves enough room to course-correct before/after and redirect actions.
- It provides the right balance of creating predictability and embracing uncertainty.

When you can take the time to split a Discovery up into different phases, it's important not to lose sight of the timeline you've committed to for the sake of following a plan. Even when your project has more of an exploratory character, you shouldn't just proceed without considering how much time is reasonably spent on each phase.

Feel free to set your best guess as the first estimate for timings. Don't think of check-ins as limitations, but instead use them as constraints that will give you as the Product Manager the best information for advancing toward concrete deliverables. It's also useful to frame the scope of the project for the people you work with.

The biggest mistake I've seen teams make when planning Discovery activities is to prioritize the time to get started over the time it takes to get to a valid insight. Let me give you an example:



- At first sight, the fastest way to "talk to users" might be to step out on the streets or into your local Starbucks to pick up insights. Unless your product is incredibly general (or an actual app for buying coffee), the chances of talking to users from your specific target audience are pretty low. That is, it takes you a lot of attempts (and thereby, days and weeks) to get insights from a significant number of relevant users.
- Going through the motions of defining recruitment criteria for interviews and waiting for users to "arrive" can feel like a long time. Effective research processes should get you access to any target audience within 1-2 weeks. And so while that might be a longer time before you can start your first interview, you can be pretty certain that the insights are valid immediately.

One shortcut I like to use in practice is to over-schedule interviews with a specific target audience upfront, so you have a tight cadence for iterating and executing problem-oriented interviews or solution-testing sessions.

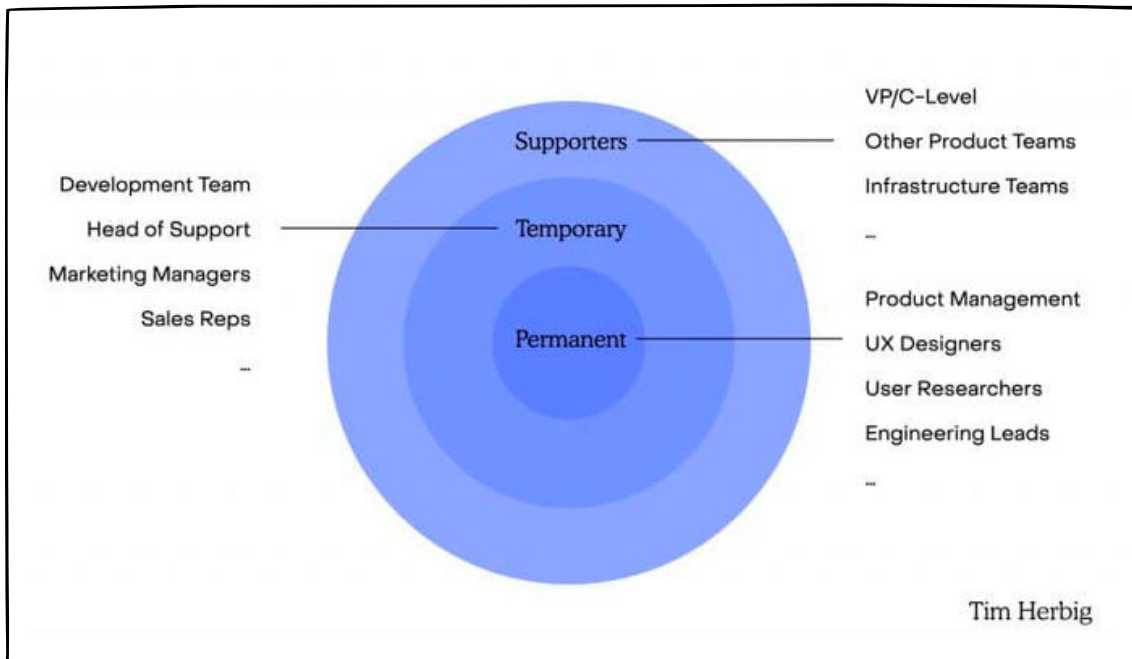
## Who Should Participate in Product Discovery?

Don't get caught up in a rigid classification of who should participate in Product Discovery. Who should be permanently or temporarily involved should be based on the requirements and context of your challenge, instead of a one-size-fits-all definition.





In my experience, it's important to identify who will be permanently involved in the Product Discovery, who will contribute temporarily, and who are general Supporters of the Mission.



The exemplary Product Discovery involvement of Product Team Members and Stakeholders should depend on context, not strict rules.

Please keep in mind that the definition of Product Discovery collaborators will and should change throughout your Discovery. The degree to which each group is involved ultimately depends on your Discovery setup. Involving fellow Product Teams during the Alignment phase might be necessary due to a high degree of dependency. And whether sales reps are valuable participants in an ideation session depends on the way you typically involve this department. And This is not a set-in-stone categorization and should allow for context-dependent adaptations.

There are two main benefits to involving team members across domains (even thinking beyond the “typical” trio of Product, Engineering, and UX) during Product Discovery:



1. The team can generate insights and make decisions at their own pace. By reducing the dependency to central departments, the Product Team can benefit from sharing their research or validation efforts. But ultimately, the Product Team owns the process and feels empowered and accountable for the results.
2. To avoid confirmation bias during interviews or ideation sessions, the holistic perspective from different domains of expertise helps to keep a balance. Each of these three main domains of expertise brings a unique set of interpretations and perspectives to the table. This way, Product Discovery efforts remain an actual team effort and results are communicated in a way that works for the whole team.

Some companies “simply” set up permanently dedicated Product Discovery teams that end up handing over their insights and results to separate Delivery teams for execution. They argue that this leads to more focus and less distraction from day-to-day work, as well as more speed and greenfield thinking. That may be true, but I believe that Product-isolated Discovery teams are a mistake. Ultimately, you should not aim to form a Product Discovery team but to make Product Discovery an integral part of YOUR team.

## What Are the Key Elements of Product Discovery?

You can think of Product Discovery in a similar way to driving on a highway toward a vaguely described destination without a GPS: you’re constantly worried about taking the wrong exit and getting lost.

Taking the first exit means prematurely committing to building an unproven solution based on non-existent problems. Waiting too long means getting lost in the Product Discovery spiral and chasing more research and testing data without ever creating value for users and the business by shipping a solution. Instead, you keep checking your environment for orientation while driving. Are you going in circles?



This analogy translates well to the non-linear, but loosely-connected, individual phases of Product Discovery teams can choose from when navigating their path toward reducing uncertainty.

### Creating Alignment to Enable Team Autonomy

In the early phases of Discovery it's important to balance alignment while exploring the problem and solution space autonomously as a Product Team.

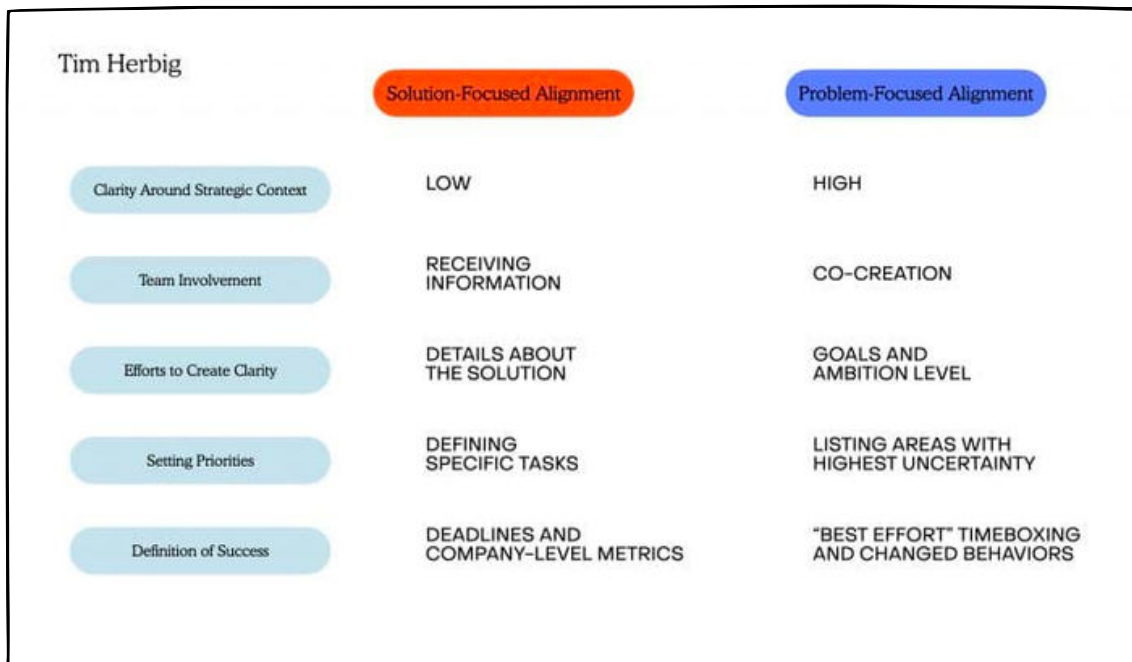
Balancing alignment to explore the problem and solution space autonomously as a Product Team is an often underestimated, yet important challenge to nail in the early phases of Discovery.

A lack of clarity around WHY you should care about this Discovery scope can lead to issues like solutioning or falling for confirmation bias down the road. But even if you're aware of the importance of alignment, not all alignment is created equal.

Almost every Product Manager knows the situation: some of your stakeholders always come up with particular features they want you to build into your product (i.e. solutions and not problems).

Unfortunately, this reveals a lack of trust: these stakeholders don't believe that your team's output will match their expectations. The best way to resolve this issue is to focus on alignment early on in Product Discovery and commit to a particular outcome rather than features.

There are other red flags I have experienced over the years to differentiate solution-focused alignment from a true focus on the problem:



The range of problem- or solution-focused attributes to watch out for during Product Discovery Alignment

Creating alignment is more than just sending an email. It also doesn't necessarily mean that you need to get everybody to agree. The more critical part of alignment is commitment. It's much easier to get commitment from your stakeholders than to get agreement. An effective way to achieve this kind of alignment together is the Mission Briefing, which we'll discuss later.

### Understanding the Most Critical User Problems to articulate real Outcomes worth focusing on

When articulating the Outcomes you want to drive, don't fall for expressing what you, as the business, want your users to do. That's a clear sign that you need more user insights and more research activities around a strategic Problem Space.

Instead, make sure you understand the problem properly in terms of the user's way of achieving their motivation. Which, when solved right, will lead to your business metrics improving as well.

There's nothing wrong with aiming to improve a revenue or conversion rate target. But that doesn't mean that your user Outcome is to "put more items in my shopping basket." That's what the business wants, not the user.



Instead, you have to understand what is in the user's way of completing the actions that will, ultimately and over time, contribute and lead to business success. Those insights might make your Outcomes read a whole lot different. Some examples could be:

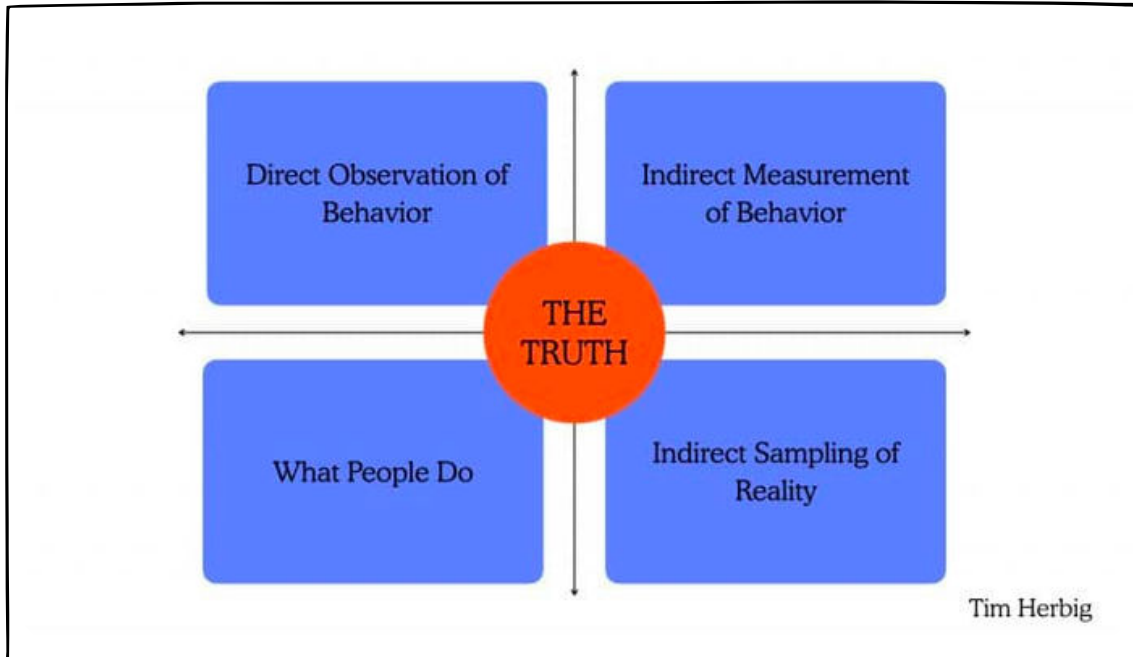
- "Continue shopping without having to leave the basket view first."
- "Compare multiple items on mobile without having to open multiple browser tabs."
- "See the delivery time implications of having multiple items in the basket, without having to continue to the checkout."

Though there's pretty much always also a business need to justify Discovery efforts, the success of your product ultimately depends on whether it can identify and solve the most pressing problems of your target audience.

When thinking of research, most teams jump immediately to interviews or send a survey. Though that isn't necessarily wrong, there are better practices to adopt for more focused navigation of the problem space:

1. Clearly state your research intent questions based on the biggest areas of uncertainty before choosing and executing a given method.
2. Make sure to collect insight from multiple angles to get to the truth that lies at the intersection of what people say and what they do.

The User Research Team at Spotify recently [shared a practical insight](#) into one of their projects. In it, Research and Data Science collaborated to look at the actual behavior of users from as many angles as possible.

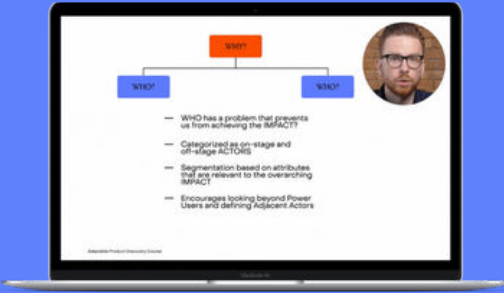


To understand the truth about current behaviors and problems, you have to choose research techniques that allow you to capture the intersection of multiple perspectives.

Remember that maybe you don't always need to engage in new research activities right away. It might make sense to turn existing sources of continuously incoming feedback into tangible evidence lockers. A convenient way to capture and structure continuously incoming customer feedback are tools like [EnjoyHQ](#) or [Airfocus](#).

Whether you're looking for a way to improve a product that is already in use or you're about to build something that aims to serve an unsolved need, identifying what your users want and need is critical.





## Deep Dive: Understanding Whose Problems Matter for Product Discovery

Learn how to determine whose needs are worth exploring during Product Discovery by using Impact Mapping in this free preview lesson from my Adaptable Product Discovery Course.

[Learn More](#)

## Collaboratively Ideating Solutions That Drive Changes in Behaviors

Once you have developed a basic understanding of which problems your users are facing, it's time to start thinking about solutions. And by that, I don't just mean to visit your CEO's pet project bucket list. If you don't want to stay stuck with incremental features that barely move the needle, you should think big.

The primary goal of the best ideation exercises is to get people out of their comfort zone and to think big(ger). Reality will catch up soon enough, but at this stage of Product Discovery let's reach for the stars.

The primary goal of the best ideation exercises is to get people out of their comfort zone and to think big(ger). Reality will catch up soon enough, but let's reach for the stars at this stage of Product Discovery.





Of course, with so many ideas being created during a lot of fun sessions, how should you prioritize them? The whole point of modern Product Discovery is to not just stay within the bubble of your Product Team, but instead involve temporary and supporting Discovery Collaborators, don't let the CEO step into the room and point to the idea they like the most. Try to democratize the process.

On the highest level, you first need to bring structure to the ideas created in the ideation sessions. Dot voting is an effective and lightweight technique to go from 40 to 10 ideas.

What about all the ideas that get lost in the process because nobody voted for them? I don't believe in idea backlogs. If you stumble upon great ideas, you should execute them, either by further exploring them or by immediately implementing them.

Great ideas will come back during future iterations and eventually find their way into the highest priority buckets. Don't worry about saving them. Instead, focus on moving forward with the ones you have selected.

## Prototyping Experiences

Putting sticky notes full of ideas into the hands of your users probably won't help you to understand their reactions. This is why you need to find a pragmatic way to turn your most promising ideas into realistic prototypes.

Though prototyping tools are the "shiny objects" of Product Discovery, you don't need to use them. Instead, aim to simulate an experience that validates the ideas in front of you. Remember that it's about prototyping an experience, which isn't always defined through a UI.



For example, let's take a look at the idea of a personality assessment test. You could prototype the experience by merely connecting several tools and creating a slim version of the product manually in the background.

You could create a basic landing page where people could start the "personality test." From there, you could lead them to a survey tool like Typeform. The answers from the survey could get pushed into whatever tool you'd like to use through a connection established by Zapier. Assuming you won't show your prototype to thousands of users, why not create the final "assessment" using predefined text blocks that match the incoming survey feedback? Combine it with a free, but professional-looking, email tool like MailChimp and your prototype is ready for testing.

With all the flexibility and power of modern No Code tools out there, it can be tempting to fully build out your idea instead of sticking to the experiment scope that's required for testing the assumptions behind it. [David Bland has a great piece of advice](#) for teams that find themselves in this position:

I still advise teams that they should defer building as long as possible, and though this hasn't changed, the sheer cost, time and capabilities needed to do a software Mashup MVP are rapidly plummeting.

Prototyping should always be a lean way to validate your hypotheses. Don't get caught up in animations or style guides. Focus on supporting the (re-)actions you want to see from users who are exposed to your product.



## Testing Discovery Assumptions Through Experiments

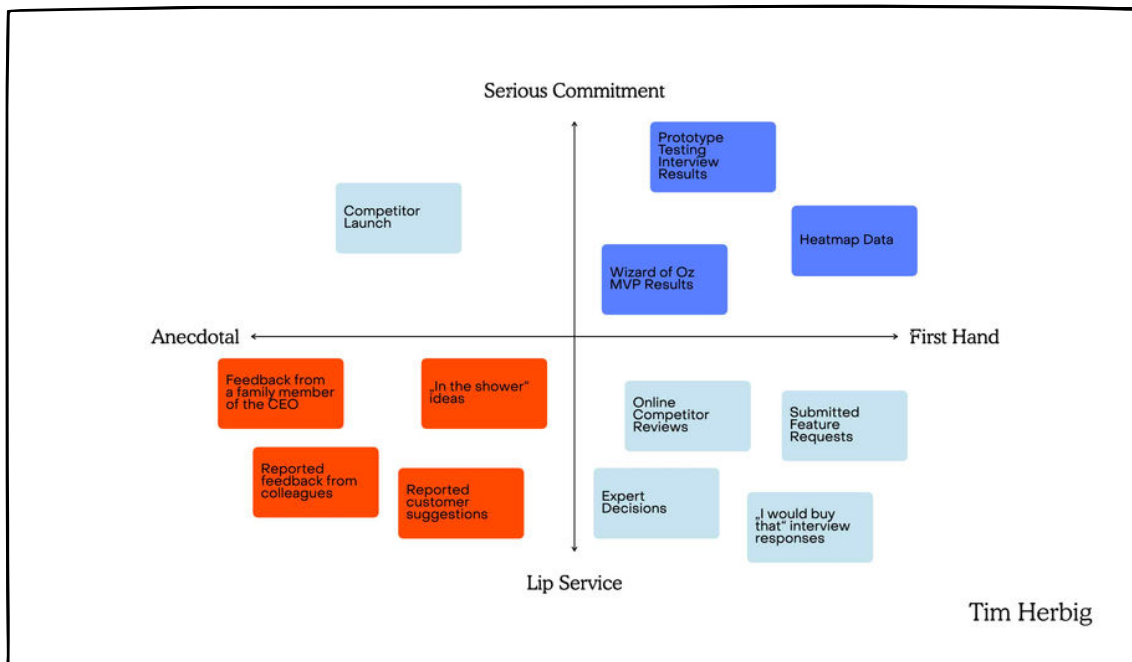
Just like Research, testing assumptions is not simply about ticking off boxes of showing a design prototype to customers or priding yourself on how many A/B tests are running on production. Instead, prioritizing assumptions and running experiments is about making sure that you collect the best information possible to evaluate a given solution based on actionable evidence.

But in order to do that, it's important to understand how relevant the pieces of evidence you collected in the past actually are.

In general, the strength or weakness of signals generated by experiments is determined by two major factors:

1. How “close” were you from the source – There’s a big difference between reported anecdotes and competitor moves where you don’t know what prompted the feedback or decision. The closer you are to collecting a signal, the more it should matter for your decision-making.
2. How serious the commitment behind the signal was – “Of course I would buy this feature” is only the most prominent expression of false desirability for an idea. Just like submitted feature requests or C-level feature suggestions based on “in the shower” moments. The more serious the commitment behind a feedback was, the more it should drive your confidence up or down.

These two attributes can be combined into a lightweight Insights Mapping structure. Most of the signals generated by typical product experiments fall into one of these four categories.



Make sure to choose an experiment technique that helps you collect strong, actionable signals, without getting lost in waiting for overly sophisticated experiment setups or anecdotal evidence.

As so often in a 2 by 2 matrix, we're looking for the stuff in the upper right quadrant and try to avoid things that would sit at the lower left.

I don't think there's a universally applicable, stack-ranked order of which signals should always trump another. Based on the decision you're looking to make and the data you have at hand, the same signal could fall into different categories. Teams don't need a universal catalog of how much each signal is worth, but a shared understanding of what they value, ie. "user interviews over competitor moves" or similar.

When it comes to selecting the actual experimental techniques to use for testing the most critical, yet unproven assumptions behind some of your ideas, you should always shoot for a holistic perspective. But make sure to list the explicit assumptions that must be true about an idea first, before selecting an experiment technique. Too many teams jump straight into the one technique they always use right away.

These definitions may come in handy:



An assumption describes a thing that is accepted as true or as certain to happen, without proof. An experiment, on the other hand, is the scientific procedure undertaken and needed to test an assumption.

Later in this guide, we will discuss techniques to help you structure your testing efforts to support experiments that truly help you develop data-based confidence in an idea.

### Refining Validated Ideas

As I mentioned in the beginning, Product Discovery is rarely a linear process. During or after the validation phase, it's likely that you need to take a step back. Whether it's "only" to pick new ideas to test from your ideation sessions, or whether you return to the initial research phase, it can be painful to go backward. But going backward is more important than moving forward just for the sake of it.

Let's say you have arrived at a set of validated assumptions and ideas that are ready for a first iteration or MVP for a real product. The work doesn't stop there. This phase of your Product Discovery process is crucial for turning your ambitions into results: it's time to prepare and transition the ideas into Product Delivery.

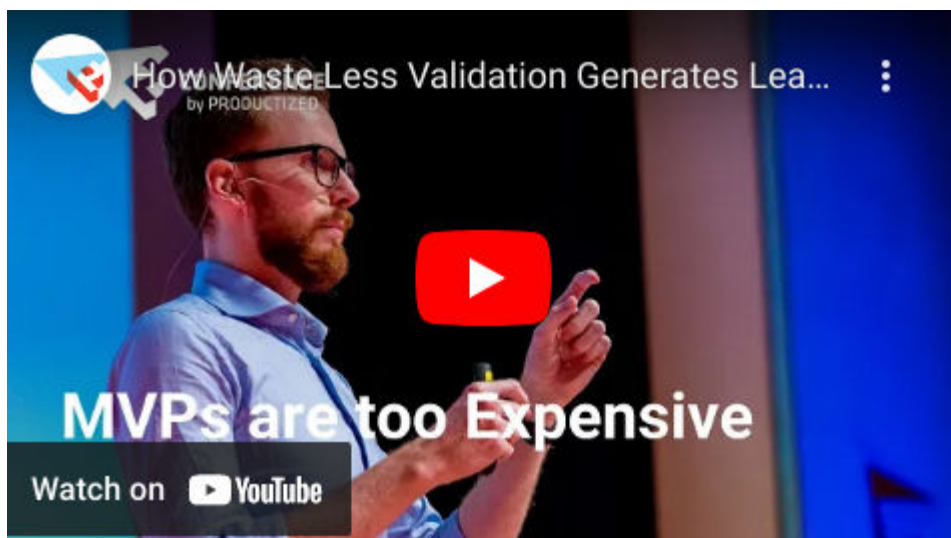
Discovery work doesn't stop after you run experiments to (in-)validate the assumptions behind a set of ideas. Product Discovery refinement is crucial for turning ambitions into results.



The concepts you used were probably pretty scrappy. In order to have something to show just in time for your latest experiment, you probably cut some corners and didn't exactly follow your company's style guide. And that's absolutely fine. But in order to avoid unnecessary conflict during the implementation of your ideas, it's now time to polish the concept.

Another challenge during the "final" phase of a Product Discovery is to slice a concept into a prioritized list of backlog items and think about potential releases. The overall vision for your product or feature idea will guide you along the way, but it shouldn't prevent you from shipping earlier iterations to learn from your users. The days of building and releasing a complex concept in one monolithic effort are over.

All too often, our first releases or MVPs are just crappy versions of all the initially planned features, released just to hit an artificial deadline. Instead, I advocate for a reduced scope MVP, which doesn't compromise on quality, but prioritizes the most valuable features.



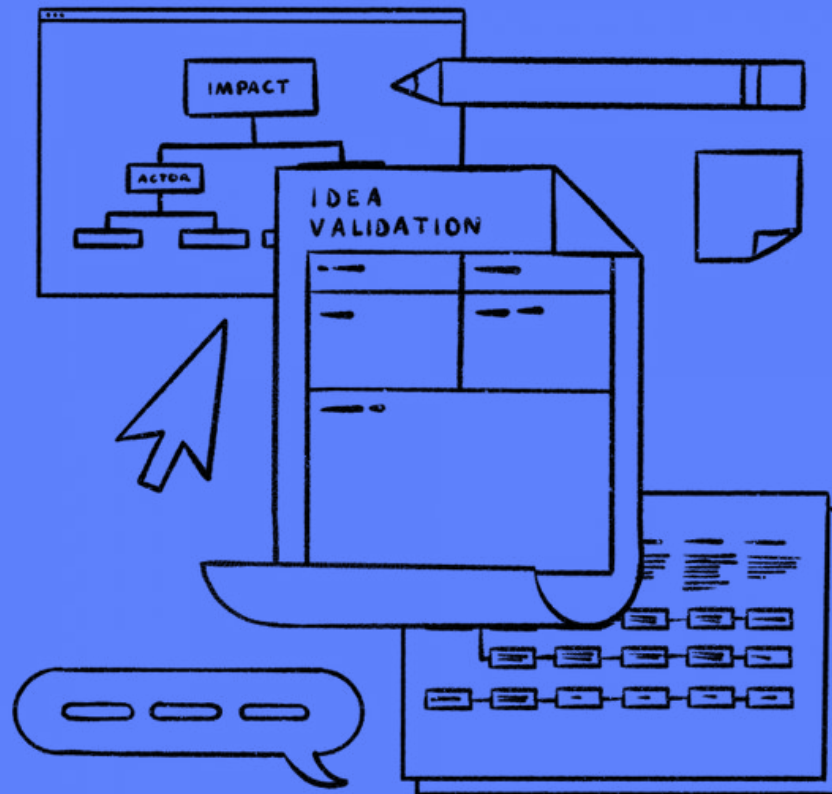


The slicing of user stories should be a collaborative effort with UX and engineering team members because the scope of user stories should be defined not only by a user perspective, but also by the necessary technical implementation.

After all, the refinement phase should focus on the very next increments you should deliver as a team. Utilize the most pressing user problems as your priority guidelines, instead of personal opinions about favorite feature ideas.



# Product Discovery Techniques and Frameworks



In this chapter, I want to share techniques that work well across the entire Product Discovery process and help teams execute with clarity and confidence.

Your Main Takeaways From This Chapter:

- How to use the Mission Briefing and Impact Mapping to structure the navigation of the problem space and solution space
- Connecting user insights to actionable Outcomes using Actor-Jobs-Outcome-Mapping.



I already mentioned my favorite probing questions about assessing a team's understanding of the problem space earlier. Since the quality of tested assumptions and implemented ideas is equally important for your product's success, here are my favorite questions to navigate the solution space:

- Which feature ideas have the most significant potential to create the changes in user behavior that will contribute to overarching business goals?

- Which assumptions behind your three most promising ideas are...

...most crucial for the eventual success of the idea

...are not based on data (yet)

- How can you test assumptions through experiments to shorten the lead time into an idea's desirability/usability/feasibility?
- What could you do within the next ten days to gain data-based confidence in the validity of an idea? Use as little written code as possible. Bonus points for not using any.

## The Mission Briefing

The strategy consultant Stephen Bungay originally described a core alignment element called the Mission Briefing in his book *The Art of Action*. In it, Bungay shows how military leaders set up their subordinates for success in the field—without narrowing their choice of options.

Strategically, these leaders couldn't micromanage their teams in the field; instead, they had to enable them to perform at their very best when their leaders were not present. Overall, Bungay's version of the Mission Briefing can be summarized into three directives:



1. Limit top-down direction to defining and communicating the intent.
2. Give individuals freedom to adjust their actions in line with that intent.
3. Allow each level to determine how they will achieve the intent of the next level up, and “back brief” so all involved parties are aware of any new changes.

Bungay’s concept has been adopted and refined by many product organizations, but here’s a practical iteration, which I adapted slightly for Product Teams. The Mission Briefing consists of five parts and unfolds its true impact on alignment when co-created by the entire team:

- The Context
- The Higher Intent
- The Team Intent
- The Key Implied Tasks
- The Boundaries

At the start, it’s important to describe the current situation from an internal and external perspective without going into interpretations or solutions. Stick to the facts. Then, close the section with the kicker, describing the problem with the current situation.

To set the context, the key is to describe the current situation from an internal and external perspective without going into interpretations or solutions.

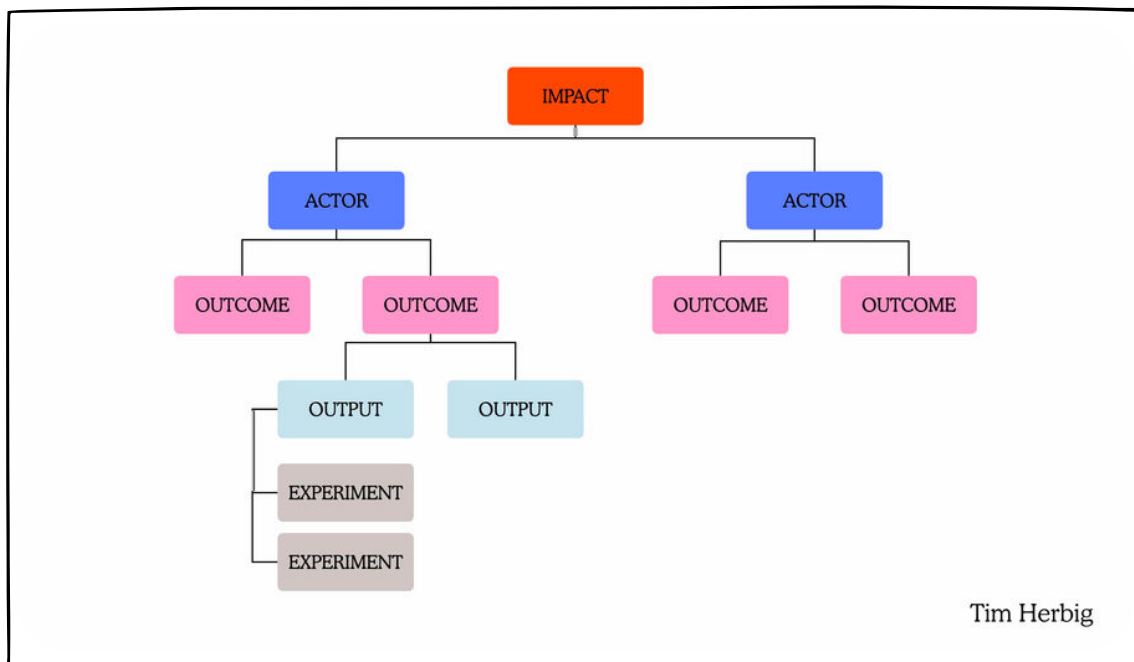
I recommend working through the Mission Briefing together section by section. It’s important to align on every section before moving on to the next one. Choices made in one section can have a dramatic impact on the other ones.



# Impact Mapping

Impact Mapping helps Product Teams connect Features and Activities to overarching business goals, using user behaviors as more leading proxies. It is one of the most effective techniques to help Product Teams make sense of all the evidence they collected and trade-off decisions. Over the years, I've iterated the framework popularized by Gojko Adzic in 2012 to specifically support Product Discovery activities and outcome-thinking.

At its core, my version of Impact Mapping consists of five levels:



The five levels of Impact Mapping based on my adaptations of the original model; connecting features and experiments to overarching business goals.

Each of these levels is associated with one of the main elements a Product Team needs to work through during Product Discovery. This makes it easier to structure the activities of teams navigating through the problem and solution space.

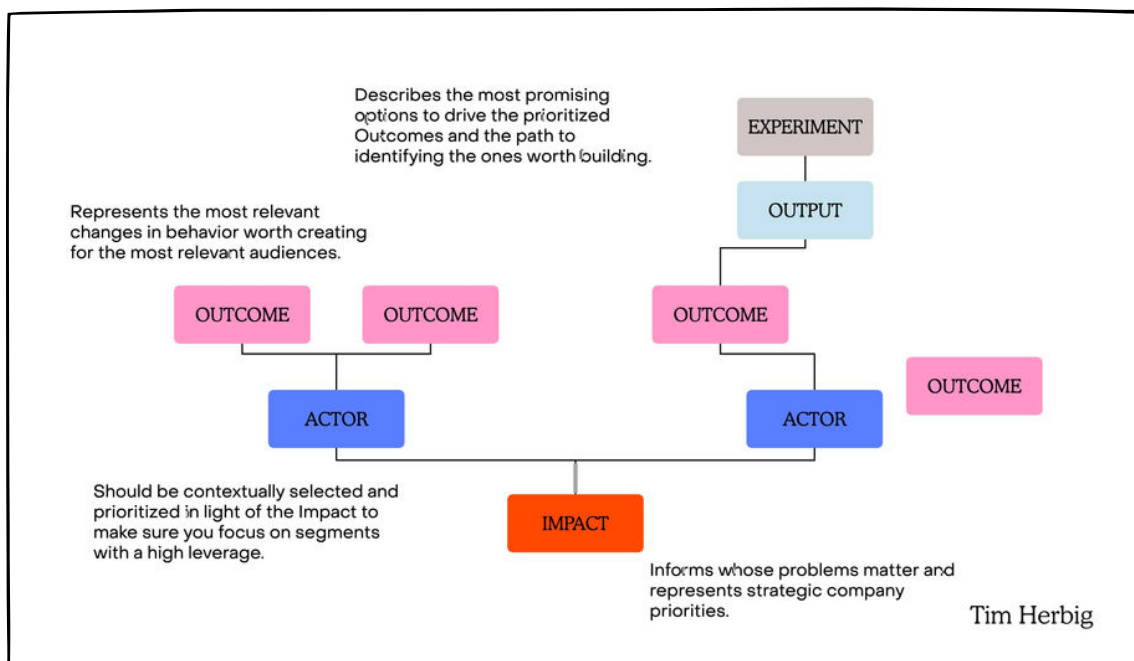
I've seen three main benefits of Impact Mapping for teams:



1. Being able to put strategic guidance, research evidence, ideation outputs, and experiment results into context to make data-informed decisions.
2. Working level-by-level, so teams (and executives) can avoid jumping straight from high-level company impacts (like increasing revenue or user activity) into pursuing solutions.
3. Facilitating tactical decisions without losing sight of the big picture of how an individual solution contributes to higher goals.

The beauty of Impact Mapping is that its levels are not just an abstract representation of an imaginary process. Instead, most of a team's Product Discovery work finds its place on an Impact Map.

Let's use Impact Mapping as a lens to emphasize how the different dots of Discovery are connected. Flipping the usual visual on its head clarifies what Impact you're contributing to or whose problems you're considering.



How the levels of Impact Mapping connect to the individual activities and insights of Product Discovery



Whether you try to contribute to an Impact around the "Average number of tweets sent per iOS user" or drive the "number of tweets containing an image" has consequences. For the latter, you'd segment actors based on their media attachment behaviors and try to understand their obstacles. This concerns the first Impact, where you'd automatically zero in on users of a given platform and try to understand their tweet frequency.

Also, Impacts and Actors will trickle down to your interpretation of insights, solution prioritization, and test the most fundamental assumptions. Because the more untested your Outcomes are, the higher the chances that your Outputs will fail.

An Impact Map can also act as a very effective input to the OKR definition of Product Teams, as it beautifully condenses insights and connects activities (or the lack thereof) to high-level company priorities. Here's a comprehensive overview of how Impact Mapping connects to and differentiates from some other popular product frameworks.

In my Adaptable Product Discovery Course, I introduce Impact Mapping as a companion guide for Product Teams to navigate through the problem space.

### Deep Dive: Using Impact Mapping to Navigate Product Discovery



Connect the Dots of Product Discovery by using Impact Mapping as a visual tool for prioritization and decision-making amidst the various inputs around the non-linear problem space and solution space.

[Learn more](#)



## The Idea Validation Grid

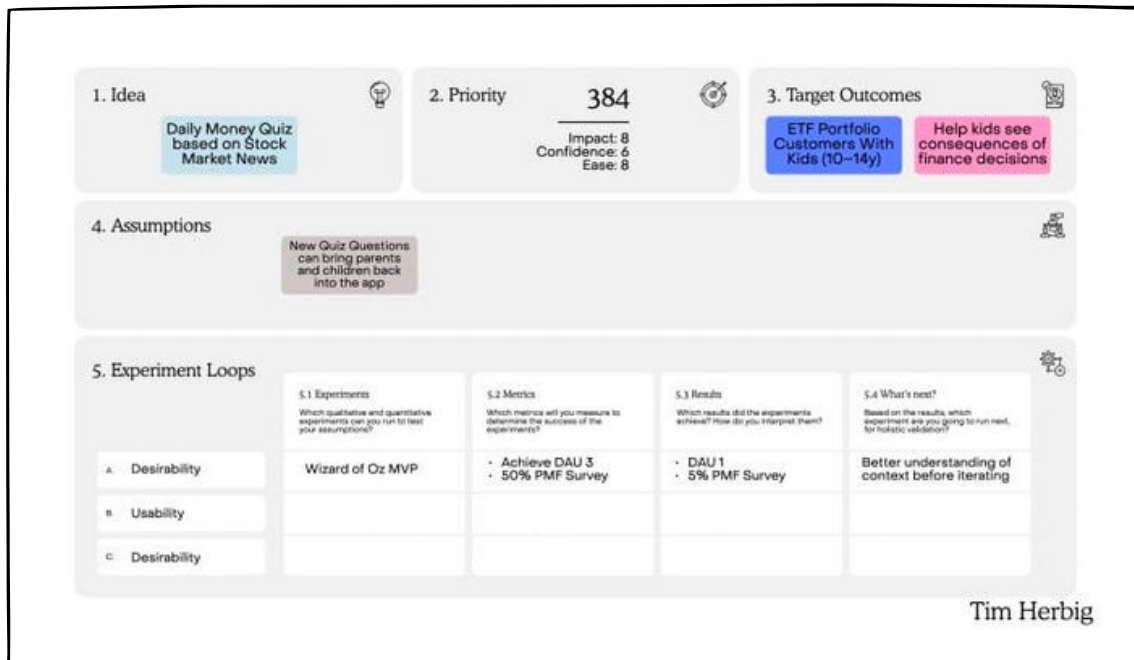
As mentioned earlier, the detailed work of planning and setting up experiments requires its own structure and place. And just as with research techniques, it doesn't make sense to jump straight to the very best validation technique your CPO has heard of. Instead, you want to build the case for why a given technique is right for the specific questions raised by your idea.

This is not just helpful for having a clear argument when discussing your resources, but it ensures high-quality discussions with other domain experts in your company about picking the right experiment. In order to do that, we need five key ingredients to create a more detailed view of our experiment design, which is inspired by [Alissa Briggs' Experiment Grid](#):

1. A high-level description of the generated idea
2. Some kind of objectified decision-making criteria like an ICE score we can use to express an idea's priority and our changing confidence in it
3. A list of prioritized assumptions about an idea (meaning, we articulate which behaviors or results the idea would need to meet in order to ultimately succeed).
4. A description of how the idea will result in a desirable change in behavior
5. The chosen experiment to test our assumptions in regard to user Outcomes as well as any next steps

Though you are, of course, welcome to find your own format for structuring these ingredients, I have found a simple structure in the form of an Idea Validation Grid to work quite well for the teams and individuals I am working with:





Structuring the testing process of Product Discovery ideas is essential for Product Teams to establish and communicate data-based confidence.

One of the key differences between the Idea Validation Grid and some other frameworks I have seen and used is a focus on the idea and the value it seeks to create, instead of focusing on the experiment. As mentioned earlier, we're not validating just for the sake of doing it. We're validating to challenge our confidence in driving user and business value through an idea.

### Actor-Job-Outcome-Mapping

During our research, we get exposed to direct insights into our user's behavior. We hear about feature requests, and, when we do our job, we get to understand problems and underlying motivations. But though all of these are valuable insights, there's extra work we need to do to make them actionable.



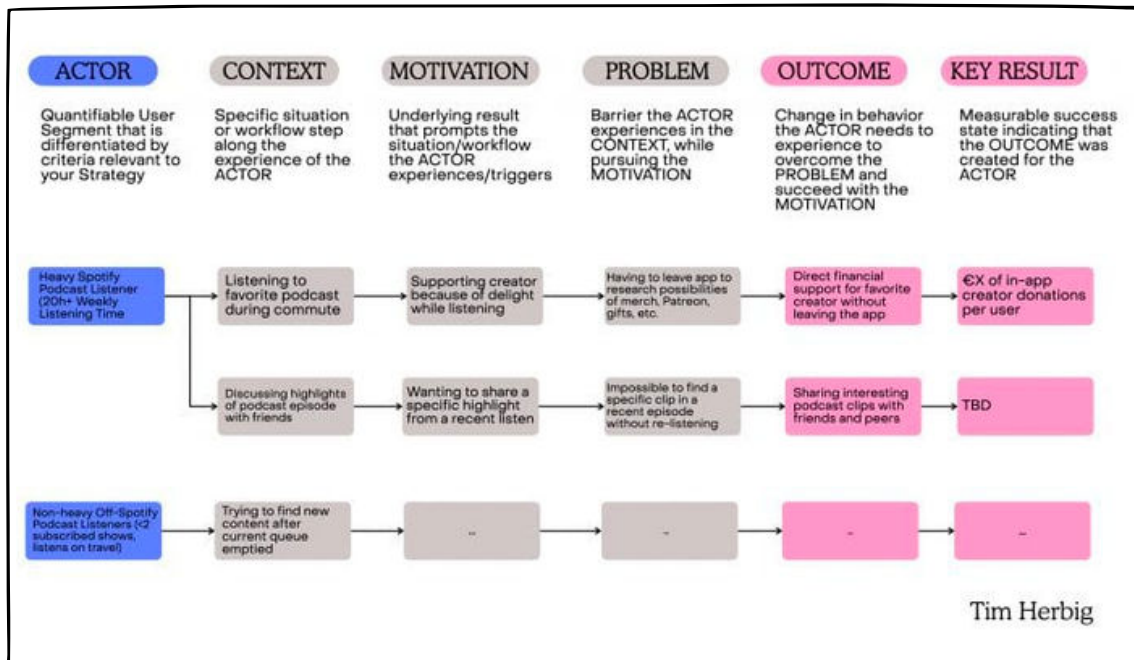
We hear about feature quests, and, when we do our job, we get to understand problems and underlying motivations. But though all of these are valuable insights, there's extra work we need to do to make them actionable.

This is the act of interpreting the insights, or articulating the changes in behavior we seek to create in the form of Outcomes. Outcomes serve as the measurable proxies we can use to determine whether we have both solved a real user problem and contributed to overarching business priorities.

As I mentioned before, your users' experiences and problems won't unfold in a linear way. So, though the components of synthesizing insights are based on a jobs-oriented approach, you might not have all the insights at hand right away. A different mapping of these components can help you understand the insights you do have as well as the gaps you have left to fill.

In addition to being aware of the core components (Actor, Context, Motivation, Problem, and Outcome), it's important to understand where to place the insights you already have and how they connect.

An effective way to communicate synthesized insights and specific Outcomes that are worth focusing on, is a structure I describe as Actor-Job-Outcome-Mapping. A connection of existing Impact Mapping elements and connecting the ACTOR and OUTCOME levels through jobs-inspired insights.



Connecting the Dots From Research Insights to Actionable Outcomes Worth Changing and Focusing on Through Actor-Job-Outcome-Mapping

Maybe you are aware of the particular Motivations and Problems an Actor experiences in a given situation. But what about other Problems that prevent the Actor from achieving the same Motivation? Or are there entirely new Motivations that you have overlooked during your first attempt to dig deeper into a specific Context? Maybe there's an entirely different Context you don't know about that could help you achieve more Impact through this particular Actor.

Plus, there could be an entirely different string of insights and interpretations for one of the other relevant Actors, plotted on your Impact Map.

Though you might not always experience such a variety and one-to-many relationships among the different insights you collect, this way of mapping what you do know helps you to uncover what you don't know—and determine the most effective technique to use next.

# Discovery Practices Reality Check



Product Discovery needs to work for your team in your company and in the context of your industry. That's why this chapter wraps things up with a focus on individual starting points and misleading "best practices."

Your Main Takeaways From This Chapter:

- Why interview and experiment cadence don't tell you much about your actual progress as a team
- How to take the first steps toward a Discovery approach that works for YOU.



The concepts and examples from this guide are aimed at giving you an idea of how to get started with Product Discovery using practical examples and a broad range of options. But your real work environment might not always be ideal—and that's ok.

In fact, the main idea behind this guide is to help you to make progress in YOUR environment and ADAPT what you're learning to your everyday life.

The worst thing you can do is say, "If I can't do X, Y, or Z at exactly the right time, the whole thing is thrown off, so forget it." ANY changes will result in a more effective Product Discovery and better products that drive customer behaviors and business results. It's not an all-or-nothing proposition.

There are a couple of practices out there that get repeated so often that Product Teams adopt them without question. I think the term "invisible scripts" applies to some of these common misconceptions:

### You Don't Need to Talk to Customers EVERY Week

The idea of talking to one user every week is a great catchy prompt for companies that want or need to shift their overall attitude toward their customers. But for the practical reality of working through the problem space of Product Discovery, you might want to reconsider. There's nothing wrong per se with continued user interactions (in whatever form). But forcing a given cadence upon your Product Team can lead to many unintended consequences.

There's nothing wrong per se with continued user interactions (in whatever form). But forcing a given cadence upon your Product Team can lead to many unintended consequences.



For example, when you're exposed to the problems of one user per week, it can be tempting to jump on what you have just heard immediately. But just because a user problem exists does not mean that you need to solve it. If you're (too) focused on squeezing in that customer interview week after week, the chances are that you stop talking to the right users.

What do I mean by that? You should focus your research efforts on those user segments that are relevant to your overarching strategic goals. Though every user problem represents some kind of opportunity, the big question is whether it's the right one for you to work on right now.

What do I mean by that? You should focus your research efforts on those user segments that are relevant to your overarching strategic goals. Though every user problem represents some kind of opportunity, the big question is whether it's the right one for you to work on right now.

## Just because a user problem exists does not mean that you need to solve it.

Depending on your stakeholder and team environment, continuously incoming qualitative insights from those weekly user interviews might not actually be helpful. Instead, it could lead to questioning your choices over and over again. At some point, you need to commit and move forward with a solution. Most likely, talking to users isn't an appropriate vehicle for gaining confidence. For that, you want to utilize quantitative techniques.

Again, of course it's important to continuously seek interactions with your customers. But blindly following a given cadence also won't guarantee "success" in Product Discovery and doesn't always give you the insights you need in order to make progress.

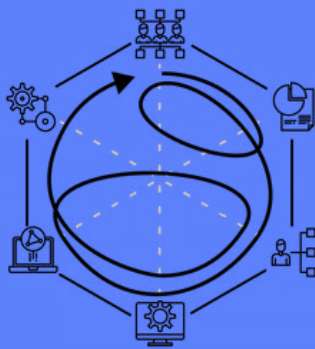


## Testing Should Focus on Insights, not the Number of Experiments

Whenever I start working with a team, department, or company, I begin by asking “how does successful Product Discovery look in your company?” The response to that question helps me further understand where an organization stands in differentiating the problem from the solution and whether the WHY behind Product Discovery is focused on P&L improvements or on a long-term belief in a fundamentally different way of working.

I often hear that “teams should run more experiments.” By itself, that doesn’t sound so bad. But when you unpack the downsides behind that statement, it becomes apparent that this is business as usual, just packaged differently.

### Deep Dive: How to Make Product Discovery Adaptable



Learn the principles and strategies that go into effective Product Discovery processes. And more importantly, discover how to use the Adaptable Product Discovery approach to make the processes work for your company and reality—right away and step by step.

[Learn more](#)



When teams embrace Product Discovery, it's often due to the fatigue of being in the hamster wheel of building more features (that nobody needs and that don't contribute to business goals). But the number of experiments conducted front and center only swaps out user stories with <insert experiment technique here>. The number of experiments conducted tells you that a team worked on tasks other than "building." But it doesn't tell you much about whether the team chose the proper experiment techniques for the challenge, executed them in a way that generated valuable insights, and even if they (in-)validated assumptions around the most promising idea.

The number of experiments conducted tells you that a team worked on tasks other than "building." But it doesn't tell you much about whether the team chose the proper experiment techniques for the challenge, executed them in a way that generated valuable insights, and even if they (in-)validated assumptions around the most promising idea.

You might argue that every experiment (if set up correctly) generates some kind of insight and COULD help the team succeed, but this often feels like sandbagging.

Takeaway: Prioritize Your Context Over Best Practices From Others

In summary, though the choice and quality of your Discovery activities are crucial, the quantity rarely is. Remember that the goal is to collect strong pieces of evidence that help you make evidence-based decisions about whether a problem is worth solving, and a solution is worth pursuing—it's not about ticking off boxes from someone else's playbook.