How to Implement Product Frameworks: Dual-Track Agile

A step-by-step guide to setting up your org and processes around the dual-track agile framework



productboard

Defining the Dual-Track Agile Framework

Agile organizations respond swiftly and effectively to changing circumstances, enabling them to thrive in a dynamic and unpredictable business environment. The primary focus is to evolve alongside customer needs. Agile development aims to help these organizations deliver value through quick product iterations—but gaps between releases persist even for the fastest teams.

Traditionally, product managers heavily relied on user feedback as the primary guide for development decisions. User input, however, isn't perfect; when feedback is housed in different places or handed off from one person to another, details get lost or misunderstood. The resource-intensive process of uncovering these insights can cause product managers to make assumptions, risking the integrity of their products. Waiting until beta testing to collect user feedback is an expensive risk if you ultimately built a feature that solved for the wrong problem.

The dual-track agile framework enables agile orgs to fill the gaps between releases with research and user-centered design. Like the name suggests, it is an agile methodology that contains two separate tracks; one for discovery and one for delivery. It allows agile organizations to run two processes in parallel—one to validate future ideas (customer value, viability, feasibility) during discovery and one to actually deliver functionality that has already gone through this validation process.

Conducting discovery and delivery concurrently increases time-to-market velocity. **As your development team works on one set of improvements, your discovery team is figuring out what the next set should be.** It takes the guesswork out of product management. While this framework is broadly applicable to whatever you're building, you might skip rigorous discovery for every little feature—especially if it's solving for a well-understood customer need.

We'll dig into how to navigate both tracks while overcoming common flaws in product discovery.

When to Use the Dual-Track Agile Framework

- You’re an agile organization that is tired of committing expensive developers to work on initiatives that aren’t properly validated
- You’re an agile organization that is looking to introduce an entirely new product or add major features to your core offering

PROS

Avoid Bottlenecks:
Waste less time uncovering new questions and risks during delivery that block progress

Increase Agility:
Quickly and efficiently make the best product improvements

Ship Customer-Validated Products:
Reduce the risk of delivering the wrong features built on faulty assumptions

Lower Development Costs:
Stop wasting effort developing features that haven't been properly validated, ultimately lowering overhead with faster dev and release cycles

CONS

Creating a separate workstream to validate the value, viability, and feasibility of ideas can require serious time investment

May also require more headcount

Anatomy of a Track

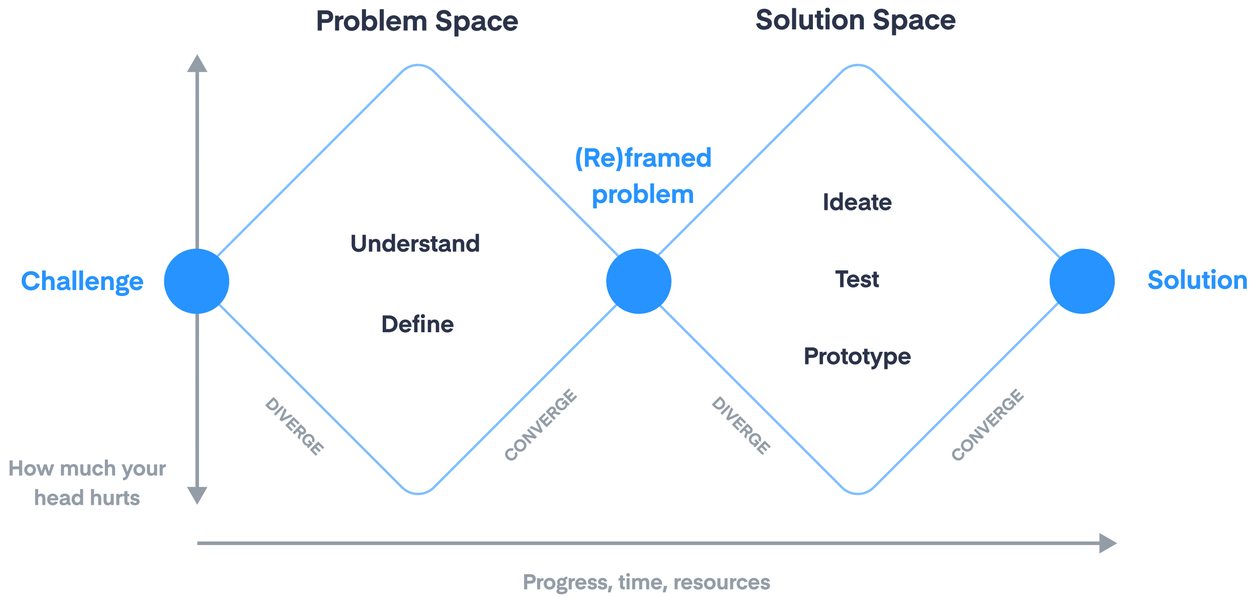
Here is everything you need to consider to balance exploration and execution.

COMPONENTS

Discovery Track

We recommend using the double diamond approach for product discovery, which requires:

- **Research and exploration**
  Understand user needs, market trends, and potential solutions through research, interviews, and other investigative methods
- **Problem definition**
  After exploring all user problems, formulate a hypothesis about the prioritized problem
- **Ideate**
  Brainstorm all potential solutions
- **Prototype**
  Generate tangible representation of your solution
- **Test**
  Evaluate and iterate before committing to a final design (A/B testing, beta testing with users, etc.)



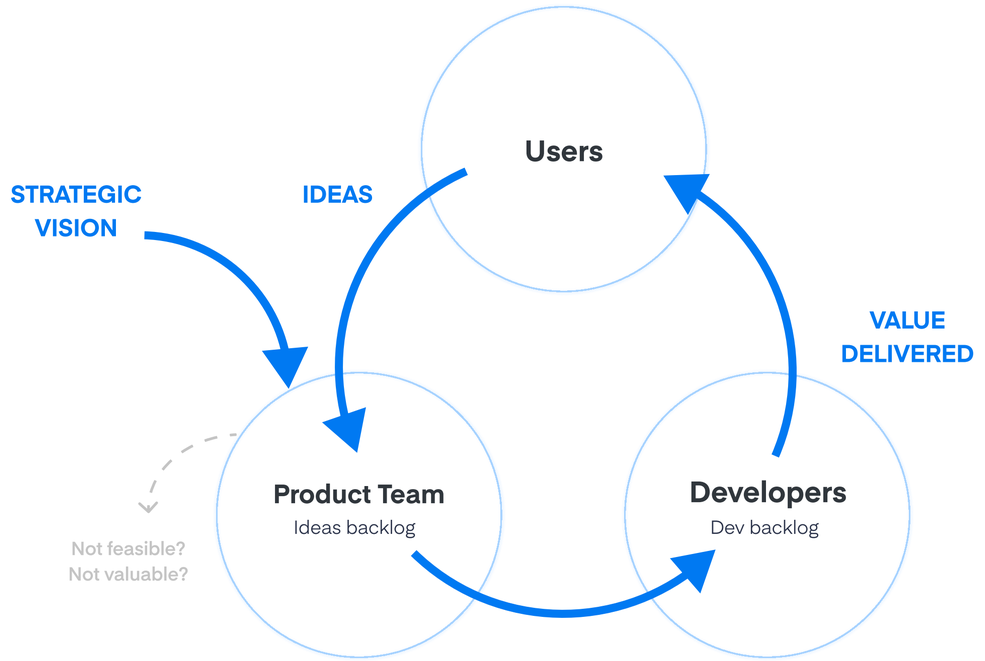
Delivery Track

Also known as agile development, this takes the learnings that the discovery track uncovered and applies them to the finished product. The aim of the delivery track is to build and release as many useful features and improvements as possible in a sprint.

- **Implementation**
  Develop and deliver features, enhancements, or products based on the insights gained from the discovery track

- **Agile development practices**
  Use agile methodologies (e.g., Scrum or Kanban) to iteratively build and release software in a flexible and collaborative manner

- **Feedback loops**
  Ensure that insights gained from user testing and usage are fed back into the development process for further refinement



ORG STRUCTURE

Successfully implementing the dual-track agile framework requires a specific organizational structure that supports cross-functional collaboration and communication between the discovery and delivery tracks.

Major Cross-Collaboration Roles

These individuals wear dual hats, serving important roles on both the discovery and delivery side. They are your glue that mends the gap that can occur between identifying what the right improvements are and actually executing on them.

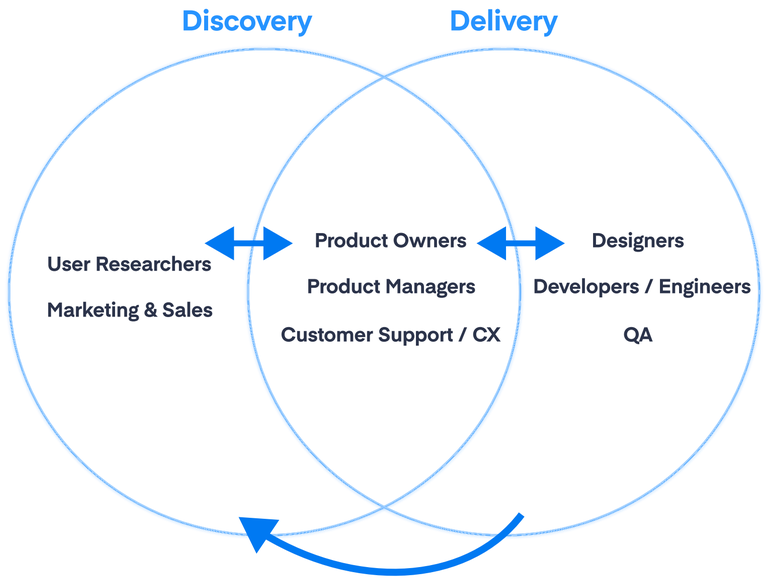
- **Product Owners**
  Acts as a bridge between the two tracks to ensure alignment and prioritize work based on user insights—the voice of the customer
- **Product Managers**
  Collaborates with the product owner (or may even be the product owner) to align both tracks with organizational goals, leveraging insights from the discovery track to inform product development decisions
- **Customer Support / CX**
  Collects valuable feedback, identifies common user issues, and understands user pain points; their ongoing relationship with end-users can lead to iterations during discovery and delivery

Discovery Stakeholders

- **User Researchers**
  Conducts studies, interviews, and experiments to gather user insights
- **Marketing & Sales**
  Encourage them to understand the value in the insights they hold and share them; go-to-market teams may hear feedback post-launch that dev and UX want to hear

Delivery Stakeholders

- **Designers**
  Translates user-centric designs, prototypes, and UX considerations into tangible features
- **Developers / Engineers**
  Responsible for coding, implementing, and delivering software features based on the requirements provided by the discovery team
- **Quality Assurance**
  Ensures the quality of the delivered features through testing to identify and fix bugs or other issues



Your Step-By-Step Implementation Guide

Let's dig into how to successfully achieve each step so you can move onto the next.

* We'll use an overarching scenario as an example: You work for a customer support SaaS company, and you are looking to implement the dual-track agile framework to ensure your strapped development team is only working on properly validated ideas.

1 Define clear tracks for your organization

First things first, you need to get all stakeholders aligned. Clearly delineating between the discovery track (focused on understanding user needs, market research, and ideation) and the delivery track (focused on implementation, development, and deployment) helps prevent redundant work.

Document everything: To clearly articulate the goals and responsibilities of both tracks, it's important to develop documentation and / or visual aids that illustrate the distinct phases and handovers between discovery and delivery. Everyone on the discovery team should know when they can move validated items to the delivery team's backlog.

Work concurrently, but don't overlap: Cross-functional stakeholders (e.g., Product Managers) should be informed about the status of both tracks and progress on these tracks should happen concurrently. But while cross-collaboration is built into this framework, make sure the approach is explained thoroughly enough that the discovery team is working towards different objectives than the delivery team.

* The dev team was excitedly discussing improving the search functionality so users can surface tickets easier... but the Customer Support team has been getting deluged with requests around reducing manual work. Right now, end-user agents are required to manually categorize and assign tickets, which is time-consuming and prone to errors. It's clear that there is a risk of delivering the wrong feature. You decide to introduce the dual-track agile framework to your agile org.

2 Assign each team member to a track

It's time to build your teams, which you cannot take lightly. A critical role is determining that of the Product Owner. Answer questions like, "*Should the Owner be the Product Manager, or is someone else a better fit?*" and "*How much time will it take each week to prioritize work and ensure everything is moving forward?*"

Don't go by department alone: We may have provided an overview on which departments best fit into the discovery and delivery phases, but you'll remember that the organizational structure is fluid and requires a lot of cross-departmental collaboration. Conduct skills assessments to understand individual strengths and assign them to the tasks where they can make the most impact. You may have one designer that loves looking through Confluence for UX feedback, and their knack for research may make them a better person to work alongside the Support team in discovery.

Account for major key competencies: At a minimum, these are the skills you should be filling for:

Discovery Track

- **User research:** Conduct in-depth user research to understand needs, pain points, and preferences
- **Prototyping:** Develop low-fidelity prototypes and validate assumptions with stakeholders and end-users
- **Market analysis:** Stay incredibly informed about industry trends, competitor products, and market demands

Delivery Track

- **Incremental development:** Break down features into smaller, incremental deliverables to enhance agility
- **Continuous integration and deployment:** Enable faster and more reliable releases
- **Iterative feedback:** Gather feedback from users and stakeholders during each iteration to make real-time adjustments

Consolidate feedback: Speaking of feedback, it's important to gather it from multiple sources to have a full-picture view during discovery—surveys, customer interviews, internal customer support teams, user testing, and reviews, to name a few. Before the discovery team dives into their research and exploration, centralizing all this scattered information can help uncover trends and eliminate access bottlenecks. This simplifies the critical process of prioritization so that those (validated) tasks are sent to delivery to be built faster.

* Team members with expertise in user research, such as the Director of Customer Success and some UX designers, are assigned to the discovery track. Their goal is to delve into customer pain points related to the current ticketing system. Developers, QA professionals, and the rest of the design team are assigned to the delivery track. They will implement only the improvements that are prioritized in their backlog after discovery.

3 Keep tracks in sync

Without communicating, how will you ensure that the insights gained in discovery are properly informing product development? And if teams aren't speaking to each other enough after the latest new feature launches, how will the delivery team even know that iterations are necessary? Even establishing just a 15-minute meeting helps in avoiding misalignment, reducing communication gaps, and fostering a cohesive approach to product development.

Establish a meeting cadence: Schedule regular sync meetings to align both tracks and share insights. Be aware: Too many meetings can harm productivity just as much as not enough. You may find yourself having check-ins three times a week to start, but don't be afraid to cancel one of the recurring syncs if it feels like everyone is just reiterating the same information.

Feedback is necessary: Establish feedback loops between discovery and delivery teams to incorporate learnings from user testing and usage into the development process. Leverage collaboration tools to keep teams connected and informed in real-time.

* Discovery and delivery meet twice a week, Mondays and Fridays, to work through the backlog. The discovery team presents insights, user feedback, and proposed features, while delivery goes over technical considerations and feasibility. There is also a sprint planning meeting that occurs at the beginning of each sprint to align on priorities for the next two weeks. The CX / Support teams have never been a part of such meetings before implementing the dual-track agile framework. Now, they are able to help inform what should be built based on their ongoing feedback sessions with customers.

4 Prioritize learning

Prioritizing learning over simply delivering features ensures that the product development process is driven by user needs and market feedback. Encouraging a mindset that values learning from both wins and losses will foster a culture of continuous improvement and innovation.

Undergo ruthless validation: Before any engineer or developer begins work, ensure that each feature implemented is based on insights gained from the discovery track.

Be ready to adapt: Use agile methodologies (e.g., Scrum or Kanban) to iteratively build and release software in a way that's scalable. You never know when user feedback will come or when market conditions change.

Consider your feedback collection strategy: Implement mechanisms for collecting user feedback at various stages of the development process—ideation, conceptualization, planning, prototyping, development, testing, launch, post-launch evaluations, and every iteration in between. Regularly review and adjust strategies based on insights gained from user interactions and market trends.

* Driven by insights from the discovery track, it has been shown that the manual categorization and assignment of tickets is the biggest problem that end-users currently face. The feature ultimately chosen for development was building an automated workflow for this task. Since launch, not only have complaints reduced, but ticket processing times have as well. While delivery was working on this critical task, discovery continued to look for improvements. Now delivery has five more tasks in its priority backlog, including that search bar update.

5 Change management

Implementing any new framework requires that all internal stakeholders are prepared for and aligned with the new direction—not just product teams.

Determine the need: Before anything, you must assess the impact of the proposed change on the people, process, and technology. Start by understanding where improvements can be made. Have one-on-one discussions with team members and ask about their work to answer questions like *"What's working well?"*, *"Where do they face challenges?"*, and *"How have you tackled problems before?"*. Don't reinvent the wheel; people often find effective solutions even without fancy tools, so make sure to uncover existing solutions before making any major investments.

Use the WIIFM communications approach: The "what's in it for me" style of communication enforces value for the change. Map out your stakeholder groups impacted by the change and assess what they will be most interested in. When introducing the change to their stakeholder group, leadership should always start with how this will benefit them.

Have a top-down approach: Change starts at the top. Executive leadership should communicate the rationale for the process change clearly and passionately.

Build your change community: Change starts at the top. While executive leadership should communicate the rationale for the process change clearly and passionately, there should also be change champions that help promote change across the company culture. When it comes to picking change champions, a complementary mix is key.

- **The educator:** They know the framework very well and can help quickly align those who don't, creating performance milestones to keep everyone excited and on track
- **The project coordinator:** They understand what the vision is for the new framework, and with their decent amount of exposure to the rest of the organization, they hold the right people accountable and routinely check in on output and status
- **The team peer:** They are closely tied to the group you want to adopt change, so it's important that they really understand the vision and the framework to become an influencer for the rest of the group

Provide (continuous) training and documentation: Supporting the transition will not be a one-off conversation. Put new / updated processes, goals, and KPIs in writing.

Follow the 5 D’s

To plan on how change will be effectively rolled out and managed long-term, you must Designate, Determine, Decide, Develop, and Decrease. (For more information on the 5 Ds, watch this change management webinar between Productboard and Joshua Childs, Director of Product Management Email Solutions at Validity.)



6 Test and validate

Develop and test your new solutions before sending them out into the market.

Validate: Create prototypes for testing with your target customers and any other key stakeholders.

Continuously iterate: Use A/B testing or gather user feedback to refine your product, making improvements based on user insights. Continuously update the software based on validation results.

Ways to Measure Success

Conducting a retrospective is a valuable practice to reflect on what worked well during your sprint, identify areas for improvement, and enhance team collaboration. Here's how to conduct a retrospective effectively:

1. **Set the stage**
   Clarify the purpose of the retrospective, emphasizing a positive and constructive atmosphere. Remind both teams that the goal is to learn and improve, not to assign blame.

2. **Gather data**
   Collect information on events, actions, or issues that occurred during the iteration. Use visual aids like a timeline or a board to document key events. Encourage team members to share their perspectives on what went well and what could be improved.

3. **Generate insights**
   Facilitate open discussion to gain insights into performance. Use techniques like "Mad, Sad, Glad" or "Start, Stop, Continue" to prompt team members to share their feelings about different aspects of the iteration. Encourage everyone to contribute, and ensure that both positive and negative aspects are discussed.

4. **Identify patterns and trends**
   Look for recurring patterns or trends that emerge. Analyze the collected data and encourage the teams to identify common themes or issues. Discuss whether certain issues are one-time occurrences or systematic problems that need attention.

5. **Generate action items**
   Develop actionable items to address areas for improvement and build on successes. Use the "SMART" criteria (Specific, Measurable, Achievable, Relevant, Time-bound) to frame action items. Assign responsibilities and timelines for implementing changes. Ensure that the actions are feasible and directly tied to the identified issues.

6. **Conduct a closing**
   Summarize the key takeaways and commit to continuous improvement. Recap the main insights and actions (e.g., email thread, Slack, etc.). Reinforce the positive aspects and express gratitude for team efforts. Emphasize that the retrospective is a recurring opportunity for growth.

Create a safe space: Team members should feel comfortable sharing their thoughts and experiences without fear of judgment. Celebrate achievements and use constructive language when discussing areas for improvement. No one should feel "called out" in front of peers.

Prioritize action items: Focus on actionable items that can lead to tangible improvements. Avoid dwelling on issues without identifying potential solutions.

Rotate facilitators: Rotate the role of the retro facilitator to encourage diverse perspectives and shared ownership of the improvement process.

Document and follow up: Keep records of retrospectives, including insights and action items. Regularly follow up on progress during subsequent retrospectives to ensure continuous improvement.

Now that we dove into what makes a successful retro, let's dig deeper into what makes a successful feature. How do you know that the dual-track agile framework is moving your org to creating more customer-centric products?

You want to see an increase in...

- ☐ Adoption rates
- ☐ Click-through rates
- ☐ Active users
- ☐ Session duration
- ☐ Number of sessions
- ☐ Retention rates
- ☐ Customer Satisfaction Scores (CSAT)
- ☐ Net Promoter Score (NPS)
- ☐ New users
- ☐ Average revenue per user (ARPU)

...but a decrease in...

- ☐ Task completion time
- ☐ Load time
- ☐ Error rates
- ☐ Churn rates

Autodesk, a global leader in design and make technology, has some incredibly complicated problems to solve. We're talking "creating an intricate system of waterways and bioswales designed to catch rainfall in New Orleans, Louisiana following Hurricane Katrina" complicated.

Given that Autodesk's Building & Infrastructure Design (BID) software products directly impact people's quality of life, their teams needed to identify, align, and act on the right problems.

That's why they have adopted the dual-track agile framework. Discovery activities help them filter out ideas they shouldn't build, while validating the ones they should. While co-creating with customers, one feature designed to improve search capability for one of the BID organization's offerings sparked an overwhelmingly positive response (over 60 notes from customers). Emphasizing the importance of validating assumptions, the foundation that dual-track agile is built on, Senior Product Manager Lilli Smith said that it was something that "surprised us."

So, that's what the delivery team prioritized building. You can learn about Autodesk's successes in being the voice of the customer here.

Helpful Tooling

Agile organizations need agile tools.

And with teams being more distributed than ever due to remote and hybrid work, there is an imperative to transition from random product releases to a more rapid and strategic market delivery. This requires a modernized approach to product management, one that can efficiently implement the dual-track agile framework virtually and at scale.

To properly act on deeply validated insights, you must be able to support cross-functional collaboration and communication between the discovery and delivery tracks. **You need a product management tool purpose-built to enable the collaborative nature that the dual-track agile framework requires.**

As the first product management platform of its kind, Productboard helps product teams conduct better product discovery and align teams throughout the development and delivery process. Productboard facilitates product discovery by providing a collaborative and organized environment for collecting, analyzing, and prioritizing ideas and user feedback. It helps teams align their product development efforts with strategic goals, ensuring that features are developed based on validated user needs and business priorities.

> See for yourself how Productboard can help support your organization's dual-track agile process. **Start a free Productboard trial today**

Productboard's customer-centric platform provides the following capabilities to support the dual-track agile framework at scale:

WITHOUT PRODUCTBOARD	WITH productboard
Without a consolidated tool to capture feedback from all sources, product discovery may not even be feasible. User research is very time consuming and manually sifting through feedback in various systems like support platforms, survey tools, and CRMs is unscalable. They're not designed to help you categorize user insights and put them to use during prioritization and delivery. This puts you at risk of missing opportunities, and you may deliver the wrong features, requiring costly rework or leading to product failure.	**Scale continuous discovery:** Capture and collect high volumes of incoming feedback in one place with Productboard's Insights capabilities. Set up workflows to route incoming feedback to the right colleagues for review. Leverage AI to identify what topics are trending, automatically surface user insights related to specific ideas, and generate summaries to understand customer needs in less time. Employ user insights to inform every prioritization decision so delivery teams can build the optimal solution.
Without a product management solution, product managers are stuck using spreadsheets or engineering issue trackers to try and decide what to build next, but these solutions lack customer data and are often disconnected from strategic objectives that should be guiding these decisions.	**Prioritize effectively:** Using Productboard's Feature boards you can easily identify the ideas that best address customer needs with auto-calculated user impact scores based on actual insights found in customer feedback. Zero in on the critical needs of your target customer segment or specific VIP customers.
Historically, product discovery was considered "complete" when product managers either began working on a feature or when the feature was delivered to customers. This does not support the dual-track agile approach of ongoing evaluation and iteration.	**Make iterations easier:** Extend product discovery post-launch, easily following up with customers after they've used a feature for the first time. This can be done through Satismeter by Productboard.
Without a strategic vision and a plan, your organization will not be aligned. Stakeholders across the organization will grow frustrated with the product plans being a "black box". Delivery teams will lack valuable context on the features they'll soon be working on. This can lead to colleagues questioning the product organization's priorities as well as costly misalignment that can delay getting new products and features to market.	**Create a source of truth for product plans:** Consolidate everything that's going into delivery with Productboard's Roadmaps that are easy to create and tailor for different audiences. Delivery teams can keep an eye on what's planned and track features as they progress through discovery. Cross-functional stakeholders can understand where the product is headed and when to expect major new features.
Without integrated tools and workflows product managers often fall back on disjointed spreadsheets or other systems that must be manually maintained alongside the delivery planning tool. Such an approach multiplies the chance of misalignment with delivery teams who may be missing important context.	**Streamline product and engineering collaboration:** Connect the entire product lifecycle from discovery through delivery. Improve productivity and alignment by integrating with your existing tech stack, including Jira and Azure DevOps. Help delivery teams access customer insights and business context behind the features they're working on.

productboard

About Productboard

Productboard is the customer-centric product management platform that helps teams get the right products to market faster. Over 6,000 companies, including Microsoft, 1-800-Contacts, and UiPath, use Productboard to understand what users need, prioritize what to build next, and rally everyone around their roadmap. With offices in San Francisco, Prague, and Vancouver, Productboard is backed by leading investors like Tiger Global Management, Index Ventures, Kleiner Perkins, Sequoia Capital, and Bessemer Venture Partners.

Learn more at productboard.com and follow @productboard.

Copyright © Productboard Inc. 2024 – All Rights Reserved