

BACKLOG

A well-prioritized agile backlog not only makes release and iteration planning easier, it broadcasts all the things your team intends to spend time on—including internal work that the customer will never notice. This helps set expectations with stakeholders and other teams, especially when they bring additional work to you, and makes engineering time a fixed asset.

What is a product backlog?

A product backlog is a prioritized list of work for the development team that is derived from the [product roadmap](#) and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first. The development team doesn't work through the backlog at the [product owner's](#) pace and the product owner isn't pushing work to the development team. Instead, the development team pulls work from the product backlog as there is capacity for it, either continually ([kanban](#)) or by iteration ([scrum](#)).

Pro Tip:

Keep everything in one issue tracker—don't use multiple systems to track bugs, requirements, and engineering work items. If it's work for the development team, keep it in a single backlog.

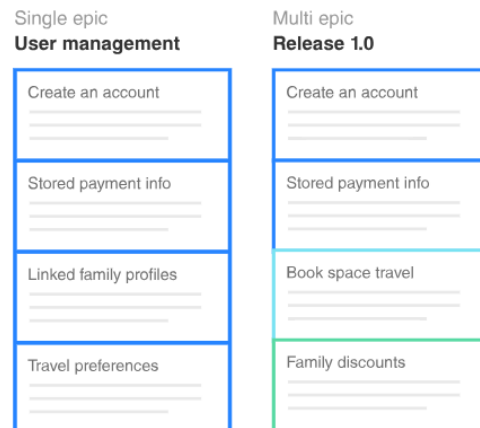
Start with the two "R"s

A team's [roadmap](#) and [requirements](#) provide the foundation for the product backlog. Roadmap initiatives break down into several epics, and each epic will have several requirements and user stories. Let's take a look at the roadmap for a fictitious product called Teams in Space.

Since the Teams in Space website is the first initiative in the roadmap, we'll want to break down that initiative into [epics](#) and user stories for each of those epics.

| User management | Travel reservations | Promos and offers |
|--|--------------------------------------|--|
| Create an account _____ _____ | Book space travel _____ _____ | Percentage discounts _____ _____ |
| Stored payment info _____ _____ | Book a hotel _____ _____ | Companion flies free _____ _____ |
| Linked family profiles _____ _____ | Book rental space _____ _____ | Customer loyalty _____ _____ |
| Travel preferences _____ _____ | Book group tickets _____ _____ | Family discounts _____ _____ |

The product owner then organizes each of the user stories into a single list for the development team. The product owner may choose to deliver a complete epic first (left). Or, it may be more important to the program to test booking a discounted flight which requires stories from several epics (right). See both examples below.



What may influence a product owner's prioritization?

- Customer priority
- Urgency of getting feedback
- Relative implementation difficulty
- Symbiotic relationships between work items (e.g. B is easier if we do A first)

While the product owner is tasked with prioritizing the backlog, it's not done in a vacuum. Effective product owners seek input and feedback from customers, designers, and the development team to optimize everyone's workload and the product delivery.

How to effectively manage a product backlog

Once the product backlog is built, it's important to regularly maintain it to keep pace with the [program](#). Product owners should review the backlog before each iteration planning meeting to ensure prioritization is correct and feedback from the last iteration has been incorporated. Regular review of the backlog is often called "backlog grooming" in agile circles (some use the term [backlog refinement](#)).

Once the backlog gets larger, product owners need to group the backlog into near-term and long-term items. Near-term items need to be fully fleshed out before they are labeled as such. This

means complete user stories have been drawn up, collaboration with design and development has been sorted out, and estimates from development have been made. Longer term items can remain a bit vague, though it's a good idea to get a rough estimate from the development team to help prioritize them. The key word here is "rough": estimates will change once the team fully understands and begins work on those longer term items.

The backlog serves as the connection between the product owner and the development team. The product owner is free to re-prioritize work in the backlog at any time due to customer feedback, refining estimates, and new requirements. Once work is in progress, though, keep changes to a minimum as they disrupt the development team and affect focus, flow, and morale.

Pro Tip:

Once the backlog grows beyond the team's long term capacity, it's okay to close issues the team will never get to. Flag those issues with a specific resolution like "out of scope" in the team's issue tracker to use for research later.

Anti-patterns to watch for

- The product owner prioritizes the backlog at the start of the project, but doesn't adjust it as feedback rolls in from developers and stakeholders.
- The team limits items on the backlog to those that are customer-facing.
- The backlog is kept as a document stored locally and shared infrequently, preventing interested parties from getting updates.

Product backlogs keep teams agile

Savvy product owners rigorously groom their program's product backlog, making it a reliable and sharable outline of the work items for a project.

[Backlogs prompt debates and choices that keep a program healthy—not everything can be top priority.](#)

Stakeholders will challenge priorities, and that's good. Fostering discussion around what's important gets everyone's priorities in sync. These discussions foster a culture of group prioritization ensuring everyone shares the same mindset on the program.

The product backlog also serves as the foundation for iteration planning. All work items should be included in the backlog: user stories, bugs, design changes, [technical debt](#), customer requests, action items from the retrospective, etc. This ensures everyone's work items are included in the overall discussion for each iteration. Team members can then make trade-offs with the product owner before starting an iteration with complete knowledge of everything that needs to be done.

Pro Tip:

Product owners dictate the priority of work items in the backlog, while the development team dictates the velocity through the backlog. This can be a tenuous relationship for new product owners who want to "push" work to the team. Learn more in our article about work-in-progress limits and flow.