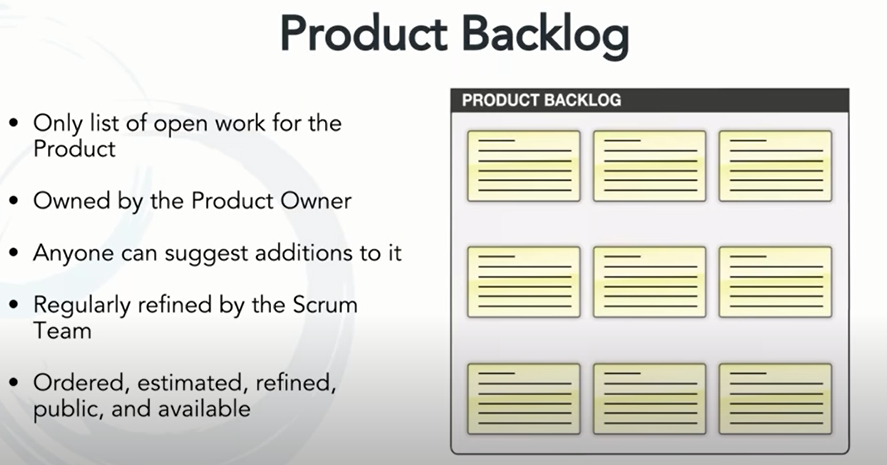
**Product Backlog**

The product backlog is a container for work you think you will do in the future to keep your product competitive. It is the output of the product owner in collaboration with stakeholders (customers, the team, analysts). It will change frequently, with items being added or taken out on a regular basis. It will be larger than the sprint backlog, generally. It will also have items with a mix of granularity; with fewer items broken down below the user story level. It is overseen by the product owner.



**Examples of Product Backlog Items**

Four main categories of items (called product backlog items) fit in the product backlog. Two are highly visible to customers — **features and bugs**. The other two, technical debt and research, are invisible to customers yet can't be ignored.

In an Agile organization, product backlog items are typically written as user stories — though they don’t always need to be. They can also be written as traditional requirements documents, or in a number of other ways.

When written as user stories, product backlog items often take the following form:

***As a <stakeholder>, I want <action> so that <benefit>.***

**1. New Features**

Requests for new features originate from a multitude of sources. These include end users, sales, support, product management, and so on. New features can be the most difficult to prioritize as you try to balance the competing needs of:

* Keeping existing customers satisfied.
* Meeting near-term sales opportunities.
* Working toward a longer-term vision of your product.

The product owner should monitor these sources routinely and resolve any conflicting requests. Doing so will help make sure the backlog contains new features that both attract new customers and build loyalty with existing ones.

***As a <new feature>, I want <to be understood and prioritized appropriately> so that <I can deliver maximum value to customers and owners>.***

**2. Technical Debt**

Technical debt includes work that needs to be done for the product to stay up to date and be maintainable. Examples of PBIs to address technical debt include upgrading to the latest third-party libraries, making architectural changes to support better scalability, or refactoring the source code to prevent future maintenance issues. When technical debt builds up – whether deliberately or unknowingly — you can risk delaying product releases.

Technical debt is often the result of change regarding:

* Direction and scope.
* Performance and scalability expectations.
* Technology or best practices.

These types of PBIs are often referred to as ”technical debt” because of their similarity to financial debt; you will have to pay interest for them, but in the form of a longer development lifecycle. These tasks should be added to the backlog, and then prioritized along with features and defects, so they can be included in the planning cycle.

***As <technical debt>, I want < to be understood and prioritized appropriately> so that <we can maintain and improve the product without delay>.***

**3. Bugs**

Bugs and defects are problems discovered by end users that escaped quality control during development. In a Waterfall process, testing is often the last step of the development lifecycle. It’s quite common to push a release live with a large collection of minor (and sometimes moderate) defects. Bugs tend to cluster and accumulate over time if they aren’t resolved. They are sometimes managed within an issue tracker, but can also be included as a part of the backlog.

***As a <bug>, I want <to be understood and prioritized appropriately> so that <problems are addressed early and the product is high quality>.***

**4. Research**

Research is another item the end user won’t recognize as a feature, but can be included in a backlog. Research is instrumental when you know very little about how to implement a new feature or concept, or want to try something new. Either way, circumstances require you set aside time to expand the team’s understanding. The output of these user stories, commonly called “spikes”, is not working code, but knowledge.

***As <research>, I want <to be understood and prioritized appropriately> so that <we can lower business risk and innovate>.***

**PROBLEM STATEMENT**

The Baby Toy Company was selling premium toys but the business had been stagnant over the years. They are looking at bringing in new offerings to increase revenue. A survey was conducted on kids and their preference for toys and it was found that attention span/engagement with a new toy lasts only a few days for kids and they start demanding a new one within a week. However, the survey also brought out that parents found affording premium toys difficult. Like any business, they saw an opportunity here and decided to launch a new offering of premium toys being available on rent and directed the IT team to make rental options for customers available online.

