Why Your Product Backlog Should Look Like an Iceberg

by Mike Cohn •

22 Comments

Image not readable or empty
/Why-Your-Product-Backlog-Should-Look-Like-an-Iceberg.png

Why Your Product Backlog Should Look Like an Iceberg

At any given time, an agile product backlog will contain a prioritized list of desired features, each sized differently and written at varying levels of detail. Because the product backlog is prioritized , the smaller, high priority items reside at the top of the list, while the larger, less urgent items fall toward the bottom.

As such, product backlogs tend to take on the shape of an iceberg, as shown in in the image below.

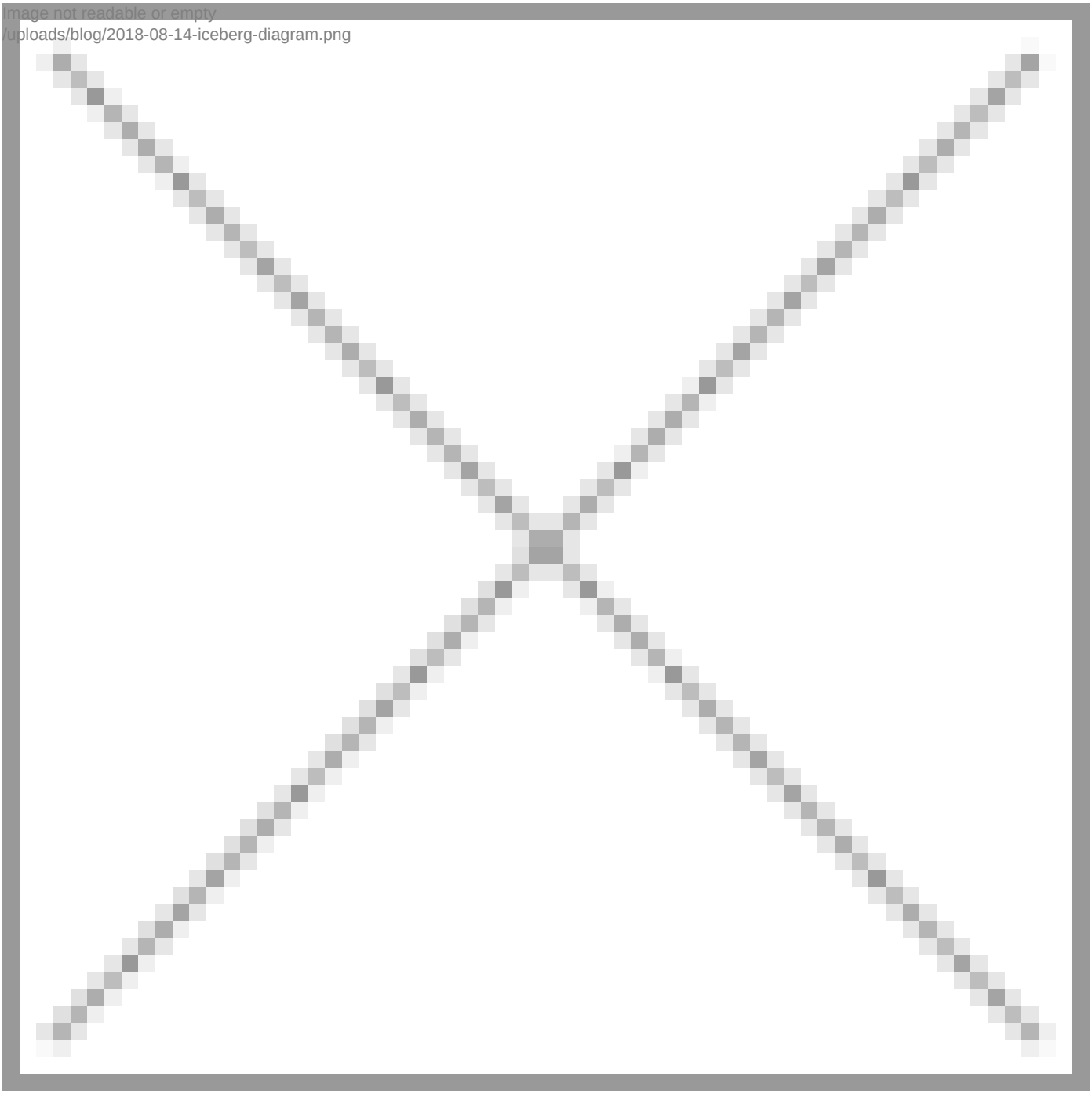
Image not readable or empty
/uploads/blog/2018-08-14-iceberg-diagram.png

At the top of the product backlog iceberg are the high-priority features the team will implement relatively soon. These items should be small and should contain sufficient detail that each can be programmed, tested, and integrated within a single sprint. As we look further down the product backlog iceberg (and therefore further into the future), items on the backlog become increasingly larger and less detailed as we approach the waterline. Teams and product owners often have only a vague idea of what lurks beneath there; some of those are features are only known in enough detail that each can be estimated approximately and prioritized.

Why Create a Product Backlog That Evolves Over Time?

It makes some people uncomfortable to have items near the waterline or beneath the surface that aren't well understood. They're used to starting a new project by identifying "all" of the requirements. However, those who are well versed in agile know that, because every project has some emergent requirements, you can never define all of the requirements upfront.

A much more agile approach to requirements is to create an iceberg-shaped product backlog that is progressively refined over time. Here's why.

Things will change.

Over the course of a project, priorities will shift. Some features that were initially thought to be important will become less so as the system is shown to potential users and customers. Other needs will be discovered and have to be properly prioritized.

If we acknowledge that change is inevitable, the advantages of structuring your product backlog like an iceberg become more apparent.

The features most likely to change are those that will be done further into the future. To account for the increased likelihood of change, these features are described only at a high level.

There's no need.

Novelist E. L. Doctorow has written that “writing a novel is like driving at night in the fog. You can only see as far as your headlights, but you can make the whole trip that way.”

Software development is the same way. My headlights don’t illuminate everything between me and the horizon because they don’t need to. They light the way far enough for me to see and respond at the speeds my car can safely travel.

The iceberg-shaped product backlog works similarly. Enough visibility is provided into upcoming items that teams see far enough into the future to avoid most issues. The faster a team goes, the further ahead in the product backlog it will need to peer.

Time is scarce.

Nearly all projects are time constrained. We want more than will fit in the time allotted. Treating all requirements as equivalent is wasteful.

With a limited supply of one of a project’s most critical resources (time), we need to be protective of it.

If it is sufficient for now to describe a future feature at a high level, this is all that should be done.

When that future feature needs to be better understood—whether because it has moved to the top of the product backlog or because we expect it to influence the implementation of another feature—we can describe it in more detail.

This is not to say that a team cannot choose to put some time into understanding items further down on the product backlog iceberg. In fact, doing so is often necessary.

If the team thinks an item further down the product backlog may have an impact on items above it, the team can put some effort into understanding it. This often results in the item being split into multiple, smaller product backlog items.

However, given our history of favoring up-front understanding of all features, teams should be careful to make sure there is a real need to better understand an item before putting more early effort into it than would otherwise be warranted based on the item's position on the product backlog.

How to Progressively Refine the Product Backlog

Now that we understand why an iceberg-shaped product backlog is desirable, let's spend some time talking about how to manage the product backlog.

As high-priority items are brought into sprints for development, they are removed from the top of the product backlog iceberg. As such, the iceberg develops a flat spot and begins to lose its shape.

To counter this effect, teams and product owners must spend time reshaping the product backlog--a process known as product backlog refinement (sometimes referred to as product backlog grooming).

Product backlog refinement can be a regularly occurring, formal meeting or can happen more informally and frequently during each sprint.

A good rule of thumb seems to be that about 10 percent of the effort in each sprint should be spent refining the backlog in preparation for future sprints.

This time may come from one individual (perhaps an analyst) whose role on the team is largely focused on the backlog. Or it may represent smaller efforts coming from each team member.

Conversations about the product backlog are not limited to a single time or meeting; they can happen any time and among any team members. These conversations enable developers to understand what needs to be built.

What Happens When a Team Refines its Backlog

A few things happen as teams refine their product backlog icebergs.

First, through conversations with the product owner, teams ensure that the items toward the top of the backlog iceberg are well understood, small enough, and sufficiently detailed to be brought

into an upcoming sprint.

Remember, that the stories near the top are high priority, so the team and product owner know they need to be implemented in the next sprint or two.

When refining these product backlog items, teams should take care in how they define *well understood* and *sufficiently detailed*.

A good Scrum team does not need a perfect understanding of a feature before it starts working on it. Rather, at the start of the sprint, the team needs to know they have a reasonably strong chance of finishing each feature during the sprint.

Second, the team and product owner might want to look at some of the lower priority items that have come nearer to the top but are still several sprints out. These items might need to be split, rewritten, or even discarded based on what the team has learned in previous sprints.

Lastly, the team might take a look at some of the features that have risen recently above the waterline. These items, which were previously lurking below the surface, are likely to be large and lacking detail.

The team might choose one or two of these items to split into smaller, slightly more detailed stories. The stories that result from this split are likely to still be relatively low in priority, so they don't need to be sprint-ready, but they do need to contain enough detail that they can be estimated more accurately and perhaps reprioritized.

(For a great primer on this whole process, including how to split user stories, check out my video training series on Better User Stories.)

What Happens When a Team Refines its Backlog

Having an iceberg-shaped product backlog should be a goal for any agile team. If you structure your backlog to have small, ready-to-develop items near the top and larger items with less clarity below you'll find your team spending less time overall in product backlog refinement. And the time they do spend will be invested in the items considered most important by the product owner.

What's Your Experience?

What does your product backlog look like? Have you experienced any of the advantages I've described? Please share your thoughts in the comments below.

Posted: August 14, 2018

Tagged: product owner, product backlog

About the Author

Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of User Stories Applied for Agile Software Development, Agile Estimating and Planning, and Succeeding with Agile as well as the Better User Stories video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also have Mike email you a short tip each week.
