

# What Does It Mean to Be Potentially Releasable?

by Mike Cohn •

18 Comments

Image not readable or empty

## What Does It Mean to Be Potentially Releasable?

The goal for any Scrum or agile team is the same: develop a potentially releasable product increment by the end of each sprint.

Sometimes referred to as potentially *shippable* rather than potentially *releasable*, the intent of this goal is for the team to have produced something good enough to be placed in the hands of users every few weeks.

But what exactly does it mean for a product to be potentially releasable?

## Just Because It Could Be Released Doesn't Mean It Should

First, it's important to understand that there is a reason it's called *potentially* releasable. The decision to *actually* release a product is an economic one. It will usually be driven by the cost of delivering the new increment of the product and the cost to customers of acquiring it.

If there are economies of scale in releasing your product, you'll want to release less often than every sprint. For example, back in the day when software products were shipped on CDs, releasing once a year or so was quite common. It was a major expense to manufacture and mail all the CDs necessary for a large product.

Similarly, if acquiring a new release of a product is expensive in time or money to customers, you'll want to release it less often. Suppose you buy the latest iPhone and a week later Apple announces a new version. And another the week after that. And then yet another the next week. You're likely to be a little disappointed that you're two generations out of date in only two weeks.

The decision about when to release is an economic one. But even you actually release infrequently, your team should still target being *potentially* releasable each sprint.

## Potentially Releasable Products Share 3e Key Characteristics

In order for a product to be potentially releasable, the improved product increment must be:

- High quality
- Well tested
- Complete

### Well Tested

Anything potentially releasable must be well tested. Not all products need to be tested to the same level--a medically regulated device warrants more testing than a web rewrite of Minesweeper. But, your customers will not be happy if you release something that has not been well tested.

### High Quality

Beyond having been well tested, the team must fix enough of the bugs so that users consider the product to be high quality. This doesn't mean every defect has been fixed. But the team should fix bugs that are beyond its [policy](#) for severity and frequency.

### Complete

For a product backlog item to be *complete*, any required associated work must be included. For example, if a product includes a user's guide, the user guide has been updated.

## But Potentially Releasable Products Don't Need to Be Cohesive

So, to be considered potentially releasable, the product increment must be high quality, well tested, and complete. But it does not necessarily need to be a *cohesive* set of functionality.

Remember creating product that is potentially releasable does not mean you must ship it every sprint. This means that functionality that needs to be included before releasing could be left out of an earlier sprint.

For example, you probably wouldn't release a system that allowed users to log in but did not allow them to log out. However, logging in without logging out could be potentially shippable if it met our three criteria above.

What a potentially releasable product does, it must do well. But it does not need to do everything. This is the heart of iterative and incremental development: An initial version of a feature is developed and then revisited and made better by adding to it.

## What Happens If You Can't Achieve Potentially Release Some Sprints

But what should a team do if it cannot achieve potentially shippable by the end of some sprints?

That depends on whether the team decides that at the start of the sprint (for example, during a [sprint planning meeting](#)) or discovers it well after the sprint is underway.

### If You Discover This During Sprint Planning

If during sprint planning the team decides it cannot reach a potentially releasable state, team members should consider dropping functionality from the sprint. Instead, perhaps, of working on seven product backlog items and not being potentially releasable could the team instead fully deliver six?

Sometimes, however, a team decides it absolutely cannot reach potentially releasable even after jettisoning some of the work from the plan. Many times this is the inexperience of the team coming through, but other times it is possible the work is just that inseparable.

If you find yourself in that situation, try a little harder before giving up on the goal of potentially

releasable. Then accept that you won't make it that sprint and proceed with the sprint as planned.

If you can't do it, you can't do it. But do try hard (and then harder) before giving up.

Finally, feel a little guilty so you don't do this too often.

## If You Discover This During the Sprint

If the team discovers during the sprint that it will not make it all the way to being potentially releasable, team members' best recourse is to discuss what can be dropped from the sprint to become potentially releasable.

Usually there is some user story that can be partially implemented or removed to make this possible. Here's where it is useful to remember that potentially releasable can be achieved without delivering a cohesive set of functionality.

## Becoming Potentially Releasable Is a Good Habit

In my experience, reaching a potentially releasable state as often as possible is simply a good habit for Scrum teams. Teams that miss reaching this standard often begin to do so more frequently or for prolonged periods.

As I've pointed out, teams will have some sprints in which they don't achieve potentially releasable. But when a team doesn't achieve that for two, three, or more sprints in a row, the team is very possibly drifting quite far from being truly releasable.

Not only does this mean the team may be delivering less value than it could, it also makes it hard to gauge how much work will be required to bring the product back to being potentially releasable.

## What's Your Experience?

Has your team ever struggled with reaching potentially releasable? How have you helped team members become potentially releasable more consistently? Please share your thoughts in the comments below.

---

Posted: September 18, 2018

**Tagged:** sprinting, definition of done

---

## 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](mailto:hello@mountaingoatsoftware.com). If you want to succeed with agile, you can also have Mike email you a short tip each week.

---