

# The Importance of Overlapping Work in Agile

by Mike Cohn • 33 Comments

## The Importance of Overlapping Work in Agile

A central tenet of agile is to avoid working in phases. On an agile project, there is no analysis phase followed by a design phase followed by a coding phase and ultimately a testing phase. Instead work overlaps in what is commonly called *concurrent engineering*.

For example, as a user interface is being designed by one team member, a second team member begins coding that functionality. To avoid rework, the programmer will start by working on parts of that functionality that appear unlikely to be affected by pending design work.

## Teams Must Become Comfortable with Uncertainty

For concurrent engineering to work, the team must become comfortable with uncertainty. For coding to begin before design is finished, the team needs to become willing to work around open issues.

Yes, before a product backlog item can be finished, those open issues will need to be decided and actions taken. But work can begin on a typical product backlog item without all uncertainty removed.

As an example, suppose a team is working on a [user story](#) of “As a user, I am logged out after n minutes of inactivity.” Before that story can be considered complete, someone is going to need to decide how long n is—30 minutes? 12 hours? But, a programmer could absolutely begin work on that story without the answer.

## An Answer Is Needed but It's Not Needed Yet

The key thing for team members to understand is that while they eventually need an answer, they do not always need the answer before starting. Some things can be resolved during the

sprint, such as the amount of time before a user is timed out due to inactivity.

Once team members fully grasp that some answers can be learned during the sprint, they become much more willing to live with the uncertainty that is needed to practice the overlapping work of concurrent engineering.

## Team Members Start Separately But Finish Together

It doesn't so much matter when team members start work on a product backlog item. What matters is that all should finish together, or as close to it as practical. In a ten-day sprint, a programmer may start work on a user story on day six and a tester on day eight. Their goal is then to finish together on day ten.

I like to equate this to running a race around a 400-meter track as in the Olympics. Because outside lanes are progressively longer, runners in those lanes begin the race further ahead physically on the track. This ensures that each person runs the same distance and that they finish at the same place, making it possible to judge the winner.

## Your Designer Is Probably in an Outside Track

An agile team can think of its analyst and designer being in the outside tracks. They need a bit of a head start. (Yes, I know in a real running race everyone starts running at the same time. But the starting line for the outside track is "ahead" of the inside track.)

The analyst needs to identify what's even being built. And the designer may need to provide wireframes, mockups or similar starting points to the team if the team is to have any chance to build a test the feature within a sprint.

When we think of the analyst and designer on the outside tracks, we can see how their head start allows everyone on a team to reach the finish line at the same time.

## How Far Ahead Does Anyone Start?

No one should start any further ahead than necessary. Remember that the goal is to overlap

work in concurrent engineering.

I am not saying that a designer is working apart from the team or working three sprints ahead. The designer is absolutely on the team and the designer's primary responsibility is helping the team in any way possible to finish the committed work of the current sprint. But even doing that, most designers will likely have some time to look ahead at what is coming next.

This is no different than what a good product owner does. A team is in trouble if the product owner is buried by the work of the current sprint, such that the team arrives at the next planning meeting without the product owner having given any thought to what to work on next.

Certain roles on an agile team need to be looking somewhat ahead. They should look ahead only as far as necessary, but some peering forward by product owners, analysts, or designers can be very helpful.

The goal should be to start the sprint with just enough information that the team can barely finish each product backlog item. Team members should feel that if they had to resolve even one more open issue during the sprint, they could not have finished that product backlog item.

## How Do You Do It?

How does your team achieve the goal of overlapping work through concurrent engineering? Please share your thoughts in the comments section below.

---

Posted: December 19, 2017

**Tagged:** teamwork, ui design, design, phases

---

