

WHY DO WE HAVE SUCH SHORT SPRINTS?



The shorter our sprints are, the faster we get the crucial feedback that is incorporated into our next step—this is how we make our development low-risk or even risk-free. Explaining the value of short sprints can be an issue, though, because not all of the examples used to explain Scrum practices in application development translate well to the mainframe.

Let's assume we are building a new application. Perhaps a sales-related app. During sprint review, the team has demoed a working "Home" screen which allows a sales officer to see their pipeline. The screen shows all of the money the sales officer has *already* made this week (closed sales) as well as all of the money that they could *potentially* make this week (pending sales) combined in the same pipeline. Although this feature was designed based on previously agreed upon requirements, the feedback now is that there is not much value in seeing closed sales and the officer would like to focus only on what is pending so that they can plan their work accordingly. Modifications will now have to be made during the next sprint.

This example not only explains the value of soliciting feedback early and often, but more importantly, shows how this feedback can directly impact or direct the next step. And while the [plans for the very next sprint](#) have changed, it's awesome that this feedback was received so early in the overall process, eliminating the need to add business logic on screens that will no longer be needed. This saves time and money, for sure! Also, this example shows why it is a bad idea to [spend months designing the entire application](#) when things could change drastically after the very first review.

For mainframe software development, however, we might have to rethink these examples. On the mainframe, there is a greater chance that the team is working with an inherited product. And while [feedback](#) is still solicited early & often (sprintly!), it might not have as much of a direct impact on the

immediate next step, for a variety of reasons:

1. While that particular story might need to go back to design/refinement, there are probably many other lines of work/features that can still proceed according to earlier plans. This feedback will not change those plans
2. Unlike the sales application in our example, changing something in mainframe code could necessitate a longer than usual revisit of the overall design. While the team is likely agile and willing to change course, the changes will not come as quickly and might even need to be prioritized behind some already existing work.
3. Mainframe teams could observe that the product owner has potentially already created future sprints fully populated with stories/cards for modifications of code that already exists and is not impacted by this feedback.

And so, when combining all of the above scenarios with the usual issues of low team capacity, high workload, competing priorities, and more, mainframe developers could feel like there is not much value in the outcome of a sprint review, thinking that the meeting is useless and wondering why it is held every other week instead of once every 6 weeks. This attitude isn't because of the Scrum practice itself, however, but because of day-to-day issues such as corporate processes and practices which do not align with [Scrum principles](#). The feelings are valid and there is no harm in experimenting with different sprint lengths to see which proves to be most productive for a team and its stakeholders.

It is important to remember that the reason we are modifying existing code is because there is an appetite for this change. We have someone paying to make this change happen. That someone is our customer, and the Agile mindset is simply about having a [closer relationship with our customers](#), seeking their feedback early and often, and incorporating that into our plans as frequently as possible.

We could be struggling with some anti- or [non-agile mindsets](#) in our workplace today, but as long as we stay true to our promise of transparency and continuous progress to all stakeholders, we can find value in shorter sprints and frequent reviews.