

NOBODY LIKES AGILE RETROSPECTIVE MEETINGS



Overview: A Product Owner and Scrum Master discuss the value of Agile Retrospective meetings and provide helpful tips for making them more effective.

Many will say they don't like Agile Retrospective meetings. They are probably the least favorite of the Agile ceremonies. You are there with your peers and discussing things that went wrong in the last Sprint. We never like to face failures, especially our own. Yet, this could be why the Retrospective meeting is the most important meeting in Agile Scrum.

We should feel a responsibility to others on the team. We should never want to let them down, yet we will. That's just the way it is, none of us are perfect. It isn't personal, in fact, we can't let it be personal. If we do, we will try to hide things and not really discuss the issues that are holding us back. It's about the functioning of the team and more so about procedure.

The meeting is not about failures—it's about improving the team. Equally important as examining problems is to look at successes. Bring up something that happened in the last Sprint that shows improvement, that an earlier issue has been rectified. Show the progress you are making. This helps shift the dynamic in the right way.

Scrum Master Perspective

Sometimes in sports there is that team that had a bad half. Things did not come together, and they are behind and heading for a loss. But after half time they come out and look like a different team—and they go on to win. What happened? It was their Retrospective. They took an honest appraisal of what went wrong in the first half and came up with better ways to work together. They

may have prepared to face this team with a set of assumptions, but they learned what worked and what didn't. They adjusted their strategy with what they learned in the first half and use that in the next half to improve and win.

As a Scrum Master, this is how I see the place of a Retrospective in Sprints. It's an important time to pause between Sprints, review what is working, what isn't and adjust for better performance in the next Sprint. Great teams, winning teams, learn from experience and know that plans need to change in the real world. This only happens with the willingness to be honest and open and most importantly a willingness to adapt and change to improve.

Product Owner Perspective

As a Product Owner my goal is to be able to deliver quality solutions to users, quickly. Friction within the process slows that down. I want to make sure the team works as efficiently as possible. That means my Stories must be clear, sized correctly and fit in the proper order in the Sprints. The Retrospective gives me insights into how my Stories have been used, and where I can make corrections to improve in the next Sprints.

So, how do you get maximum value out of your Retrospectives? Follow these steps:

1. Focus on the goal of being a better functioning team with fewer issues.
2. Focus on procedure, not personalities.
3. Focus on root causes, not symptoms.
4. Look for and suggest solutions; don't dwell on the issue itself longer than necessary, but keep at it until you come up with possible solutions.
5. Keep a record--put down what changes you will make to improve, and then in the next meeting, go back to that and see how it's going.
6. Explain why something impacted the team because the issue alone may not stand out; be detailed in describing how the team's work was affected.
7. Encourage the team to make a quick note of any items they want to discuss throughout the Sprint--and require each team member to contribute.
8. The Scrum Master should ideally facilitate the conversation, respecting everyone's viewpoint while remaining neutral. And never turn it into a lecture!
9. Maintain a simple doc of what was discussed and make it accessible to the entire team.
10. Lastly, but probably most importantly, not everything is a problem. Share what went well! Don't overlook successes especially for newly implemented solutions.

The Retrospective meeting is probably the most critical meeting if we intend to improve. Without allocating the time to specifically talk about improvements, how could progress happen?

Originally published in *Enterprise Executive*.