



Agile Retrospectives

A retrospective is anytime your team reflects on the past to improve the future. Between technical and non-technical teams, you can retro on just about anything! Right now, we're hosting a public retrospective on agile software development. Help define the future of agile by adding some of your ideas to our board.

Join the #RetroOnAgile conversation

Reflect on your software development by tweeting with #RetroOnAgile. Tell us one thing you like with #ILike, one thing you wish you could improve with #IWish, and one thing you'd like to see in the future with #WhatIf. Use the hundreds of responses below as inspiration. Your feedback will show up here within 24 hours.

Join the conversation

Pro Tip: ^^^ Replace the question with your answer, leave the hashtags ;-)

Our #RetroOnAgile board

I LIKE



Up Next
Kanban →



Why run a retrospective?

In 2001, with the stroke of a pen, the agile retrospective was born. The last of [the twelve principles of agile development](#) reads as follows:

"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."

The agile manifesto makes it clear: In order to best live the agile values, teams should meet regularly to check in and make adjustments. Most commonly, development teams apply this principle by hosting regular retrospective meetings, and while that meeting is the focus of much of this page, it's not the only way to retro.

More recently, the concept of retrospectives has made its way out of development teams and into all facets of business and teamwork.

I know marketing teams that retro on campaigns, management teams that retro on large presentations, and above, Atlassian is hosting a retrospective on their entire industry. This openness to retrospectives, and their proliferation into all facets of business, is something to get incredibly excited about.



values:

- Individuals and interactions over processes and tools
- Responding to change over following a plan

At face value, this is what a retro is all about: Working with real people to make changes and improvements. Few things reinforce agile principles better. Now that we know why retrospectives are so important, read on and learn how to host a meeting on your own.

And don't forget, we do retrospectives to make things better, so if you're into agile, [participate in our #RetroOnAgile](#) and help define the future of software development.

The retrospective meeting

Retrospectives are an excellent opportunity for your agile team to evaluate itself and create a plan to address areas of improvement for the future. The retrospective embraces the ideal of [continuous improvement](#) - and protects against the pitfalls of complacency - by stepping outside the work cycle to reflect on the past:

The purpose of the retrospective meeting is to:

- Evaluate how the last sprint, iteration, or work item went, specifically around the team dynamic, processes, and tools.
- Articulate and stack rank the items that went well, and those items that did not.
- Create and implement a plan for improving the way the team does work.

The retrospective provides a safe place to focus on introspection and adaptation. In order for retrospectives to 

be successful, there needs to be a supportive atmosphere that encourages (but doesn't force) all team members to contribute.

The retrospective should be a positive, energizing experience for your team. It helps team members share important feedback, let go of frustrations, and work together to come up with solutions. Facilitators can also get a lot from the retrospective, including a better understanding of how the team works together and what challenges (and successes) they experienced in the last sprint. A successful retrospective results in a list of improvements that team members take ownership of and work toward in the next sprint.

How to run your first retrospective

While it can be beneficial to vary the format of the retrospective (more on that below!), certain aspects like timing, attendees, and general format should remain as consistent as possible.

The when:

For agile teams working in the traditional two week sprint, the retrospective should take place at the end of every sprint. For teams running a more Kanban-esque style of work, a monthly or quarterly retrospective may make more sense. It's also healthy to engage members of the broader leadership after major initiatives have been rolled out; be careful to focus not on what was delivered, but rather on how the team worked together produce it.

Plan to spend at least thirty minutes, and up to an hour, depending on how long the sprint is and how much you have to cover.

The who:

Every team member should attend the retrospective, with a facilitator leading the discussion. The facilitator can 

be the scrum master, product owner, or it can rotate throughout the team. Feel free to pull in designers, marketers, or anyone else who contributed to the current sprint or iteration.

The what:

There are several ways to mix up your retrospective (which we'll discuss below), but here's a basic template for retrospective meetings:

- 1 **Create a short list of things that worked well and things that could be improved.** This list can be created on a whiteboard, on an [Atlassian Confluence page](#), or maybe even sticky notes on the wall! No matter where you capture the initial feedback, be sure to memorialize it right after the meeting so it can be referenced down the road.
- 2 **Prioritize this list by importance as a team.** You may discover common themes, which can be grouped together.
- 3 **Discuss ways and tactics to improve the top two items on the "room for improvement" list.** Focus on outcomes, not actions or people, or the past.
- 4 **Create an action plan.** By the end of the session, the team should have produced a few actionable ideas with clear owners and due dates to address the areas of improvement.
- 5 **Be disciplined about executing #4.** Nothing is more frustrating than repeating the same roadblocks in every retro. Avoid stagnation (and frustration!) by making sure everyone walks away with clear next steps. Each action item identified in the retro should have a clear owner who follows it through to completion.

Because variety is the spice of life



Standardizing your retrospective is a good idea to create consistency and to build trust amongst the team over time. But there are a few "tweaks" facilitators can try that may help uncover additional insights, encourage participation from new team members, or simply [keep it interesting](#).

Bring in an outside facilitator. Typically retrospectives are run by the scrum Master or project lead, but you may want to consider bringing in a guest to facilitate your next retro. The dynamic may shift in a positive way by having someone with no "skin in the game" lead the discussion. Moreover, this strategy enables someone else within the organization to observe how other agile teams are working and may pick up some best practices for his or her own team.

Vary the list prompts. At the end of the day, the retrospective is meant to uncover what's working and what's not. But arriving at those xxxx may be xxxx. Consider these different prompts:

- Start / Stop / Continue: What the team should start doing, stop doing, and continue doing. Focus on ways to discontinue items in the "Stop" column.
- More / less: What the team needs to do more and less of. Create a plan around how to tackle the top items in the "do less" list.
- Glad / Sad / Mad: What made the team glad, sad, and mad. You guessed it, focus on the sad and mad lists and how to improve so there are only items in the glad column next time.

Engage leadership. After a major project has been rolled out, schedule an hour with a member of your leadership team and focus on how the team worked together (not particulars of how the initiative went).

There are [plenty of ways to improve](#), so don't hesitate to find some new tricks of your own. Whether you're trying to [engage a distributed team](#) or improve a retro process that has stagnated, the key is to keep your team engaged and make the results actionable. →

Join the conversation!

Now that you know the basics of running a retrospective, we'd love to hear from you about your team's retrospectives. Start your tweet with I like “I wish” or “What if” and you may just see your feedback on our virtual board above! [Join the conversation →](#)

[Agile](#) / Retrospectives

Subscribe

Stay abreast of the #RetroOnAgile and other agile trends

Email

[Subscribe](#)

PRODUCT FEATURED



Jira Software

The #1 software development tool used by agile teams

<https://www.atlassian.com/agile/scrum/retrospectives>



the most software development tool used by agile teams

[Try now for free →](#)

TUTORIAL

Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

[Read this tutorial →](#)

UP NEXT

How to Run an Agile Retrospective Meeting with Examples

Instructions and variations on the classic team retrospective. Popular with agile developers, but not just for software teams anymore.

[Read this article →](#)



Agile Topics

Agile project management

Scrum

Kanban

Design

Software development

Product management

Teams

Agile at scale

DevOps

Sign up for more agile articles and tutorials.

Email

Subscribe

