Table of Contents

How Do We Run Our First Sprint (or Iteration) Retrospective?

Premise

At the end of a sprint, the Team holds a Sprint (or Iteration) Retrospective to review their work process and the way the Team worked together (the “inspect and adapt” cycle for the process and people). This information can help the Team determine how to improve performance in the next Sprint / Iteration. Typically, the Retrospective is held after the Review / Demo.

Background

The Team can consider potential improvement topics, such as the following areas:

- Quality: How to reduce defects which escape from the Sprints?
- Predictability: How to bring committed velocity close to actual velocity?
- Throughput: How to increase the value of work delivered each Sprint?
- Team health: How are we working as a team?

In addition in most cases should be an energizing (if sometimes exhausting) event.

Everyone on the Team should participate in the Retrospective. Others may be invited, but this is essentially a Team-focused activity. The Team can determine who they want to invite on a iteration-by-iteration basis.

The Scrum Master's role during the Retrospective is to facilitate the Team's discussion about improving their delivery process. It may be helpful, and a beneficial change of pace, to have someone else facilitate the meeting so the Scrum Master can participate as a “normal” Team member during the Retrospective. For example, another Team's Scrum Master could serve as facilitator.

The result of a Retrospective is a prioritized list of potential improvements with a few selected to be worked on during the next Iteration. Often these improvements are defined as backlog items, to ensure that they are worked on in the Iteration. Like any backlog item these are estimated, prioritized and, when they become part of the Iteration, are detailed just like any other backlog item.

During the next Iteration's Retrospective, prior improvement items should be reviewed for progress made in achieving the expected result or to understand what happened, whether the improvement was (or still shows potential to be) successful, or whether something else needs to be tried.

As much as possible try to treat these improvements as “experiments.” In other words, we want to set up an experiment with an expected result, we understand how we are going to track the experiment (metrics) as we run it, and we then run the experiment in the next Iteration to see if the expected result occurs. There are a couple of reasons for doing it this way:

- We want to encourage a general Team level of risk taking but we want to have intentional risk. An experiment is a way of phrasing and running something risky without making it seem like an "all or nothing" approach.
- In the wider organization, we want to the "experiment" terminology to catch on as this is a mechanism to allow continuous improvement happen.

The results of the Retrospective, the experiments, are published so others (teams, management, etc) can see them. The idea is that others could review this information and learn from this team, adapting ideas to their own environment. Note that the activities and information used to generate the experiment is private to the Team. The Team needs to feel they have a safe environment to hash out issues. If this information has to be public then the team will not be able to talk about sensitive issues. So the result is public, but the process is private to the Team. In other words "we are not interested in the sausage making, but we are interested in the sausages."

Structure

Duration: Maximum 1.5 hours for a 2 week sprint (10 working days).

Who: The Team and others invited by the team.

Summary Agenda:

- Review results of last Retrospective
- Review last Iteration
  - Review metrics (such as planned vs actual work, throughput / velocity, cycle time on work)
  - What worked? What isn't working?
  - What adjustments should we make to improve performance?
  - Health check for the team
- Identify specific changes to implement and monitor in next Iteration - improvement efforts
- Is there anything we want to raise to management that would help?

Result: A couple of improvement experiments to try in the next Iterations, publish to a public place (team site).

Meeting Outline

Esther Derby and Diana Larson in "Agile Retrospectives" talks about the two loops of the inspect and adapt cycle associated with Scrum:

1. Deliverable: Culminating with the Iteration Review / Demo, the inspect and adapt cycle includes planning, building (including testing) and reviewing the product increment. This is a “plan, do, check, act” cycle for the deliverable we are producing. Each cycle can be seen as an “experiment” – “we think this is what they want, we show them what we think, we get some feedback, and we update our plan based on what we see.”
2. Methods and Teamwork: Like the deliverable we need a “plan, do, check, act” cycle for the improvements associated with the people and process we use.

This is important as for many teams the Retrospective starts and ends with the question “what do we think we need to do better” without having a complete understanding of a problem or, more importantly why we have the problem. To combat this short term approach, they suggest that the following structure should be put in place when running a Retrospective:

1. Set the stage: Get everyone to say something and check into the process.
2. Gather data: Brainstorm what happened during the Iteration. People have a tendency to remember easily what happened in the last few days of the Iteration, but often have a hazier view of the early part of the sprint. Worse people will often see the same event but have totally different views of the event. The idea is that, before we decide what experiments we are going to run, we should make sure we have a common understanding of what happened. It is important to treat this part of the Retrospective as a “no judgment” zone, so we get all the perspectives.
3. Generate insights: Look at the data we have, understand what it means and then brainstorm ideas to address these. Again, it is important to treat this part of the Retrospective as a “no judgment” zone, so we get all the ideas.
4. Decide what to do: Prioritize the ideas and decide as a Team which ones we need to take on in the next Iteration. Put together a experiment to figure out whether the idea has merit. Use SMART (Specific, Measurable, Attainable, Relevant, Timely) goals to help formulate the experiment.
5. Close: And now its time to celebrate the conclusion of another successful Iteration
6. .

Sample Retrospective

The following is a simple retrospective “script” aimed at working through your first retrospective:

- Preparation:
    - Have team metrics (actual throughput / velocity, committed vs actual throughput / velocity, cycle time, and others such as % unplanned work) displayed in the room
    - Have “retrospective prime directive” displayed in the room - see below
    - Have “timeline” drawn up in the room - see below “running retrospective” for image
    - Have “liked, didn’t like, kudos, idea” grid drawn up in the room
    - Food in the room, perhaps something to celebrate with as well
- Introduction: Talk about the reason for the retrospective (“to improve our effectiveness”), briefly discuss the approach we will take and why, and point out and discuss the “retrospective prime directive”.
- Set the stage: Ask the team (everyone to respond) “In one word, describe what you thought of the last iteration”.
- Gather data: Point out the “timeline” diagram on the wall and describe how it is used – when event

occurs on X-axis, how “good” or “bad” you thought the thing was based on distance above and below the line. Then have the team pair up and interview each other about “events that happened during the Iteration.” While one person talks about events they remembered, the other acts as a scribe writing down what they hear (one item per sticky note) only asking questions to clarify ideas (no judgment). Then have pairs switch roles. Finally have each person go up to the time line and talk about the things they remember, placing the sticky notes on the timeline and, when something is up there already that is similar, group those items together.
- Generate insights: Point out “liked, didn’t like, kudos, ideas” diagram. Explain usage – “liked” is for those things we liked that we want to continue doing or even improve, “didn’t like” is for those things we didn’t think were good as well as a couple of ideas on how we could address, “kudos” is to provide kudos to someone on the team that needs appreciation as a result of something they did to help the team, and “ideas” is for other ideas we have that are things we might want to try to improve. As joint team activity, have people either move items from the “timeline” to the “liked” board or create new ideas (more stickies) and “kudos” items. For the “improvement” areas generate a couple of ideas on experiments we could run as part of the move.
- Decide what to do: Dot-dot vote to prioritize the ideas. Pick the top 2 or 3. Then split into a couple of groups have each group generate SMART goals which define the experiment the team wants to run. Review results with whole team and finally decide which ones the team is going to take on in the next sprint.
- Close: time to celebrate.

Potential Retrospective Topics

The following are suggested questions / topics that could be used to prompt Scrum Team members to discuss possible improvements:

- Improve quality by asking:
  - “How do we get closer to 'potentially shippable'?”
  - “How do we get to zero bugs in this Iteration?”
  - “How do we reduce the technical debt of the product or service?”
  - “Have we considered things like administration, stress/performance testing, ease of installation and support?”
  - “Have we reduced the amount of time it takes to move completed work to production?”
  - “How can we get closer to the customer?”
- Improve predictability by asking:
  - “How do we get committed velocity / throughput within 10% of actual velocity / throughput?”
- Improve throughput (throughput is time to get an idea or concept into shippable code) by asking:
  - “How could we have taken on one additional Backlog item this Iteration?”
  - “How can we focus more on delivering product backlog items?”
  - “What delays could we remove from how we work so we could go ‘faster’?”
  - “How could we increase the work we do in parallel to reduce ‘serial’ handoffs”?
  - “How could we reduce the average cycle time of an individual story going through a Iteration”
  - “How could we ‘swarm’ more on the work we do in a Sprint and so increase focus and

decrease the amount of work we have in progress”
    - “How we can reduce the amount of work-in-progress (WIP) we have so that we reduce the risk associated with completing the Sprint and increase our ability to deliver”. Note: many Scrum teams have the “cliff pattern” in a Sprint where all the stories are considered “in progress” until the last day of the Sprint. This pattern means you don't have a smooth flow of work through the team.
- Improve team health by reviewing how Team interacted with each other. Some tools often will help this discussion. For example:
    - Use the Spotify Squad Health Check
    - Use the Five Dysfunctions of a Team survey to discussion team dynamics
    - Use a Myers-Briggs survey to discuss how people on the Team interact differently.

All else being equal, focus on improving the Team's ability to make and meet commitments each Iteration, e.g., eliminate waste (don't create bugs) and decrease delays.

The order – quality, predictability, throughput – of these questions is significant. Experience shows that improve quality and predictability will typically improve throughput as well whereas the reverse is not true.

Additional Ideas

The Retrospective Prime Directive

One thing we need to do is try and establish a “safe” place during the retrospective to express ideas, criticisms, problems, opinions and so on. One way to help this is to start the retrospective by reviewing the “retrospective prime directive”. This states:

> “Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand.”

In other words, lets assume “good intent” and not that someone made it their mission in life to do something bad to me. We need this because as we go through the retrospective process we will discover decisions and actions we wish we could do over. This is wisdom to be celebrated, not judgment used to embarrass.

Distributed Teams

In some cases, for example where there are big time zone issues, it may not be possible to get the whole team together for a lengthy meeting. If you do not run a Retrospective meeting, then you still need to determine a way to do incremental improvements, Iteration by Iteration. The team should be able to talk about improvement goals for the Iteration, how they are progressing those goals, and whether the experiment represented by the goal has been achieved at the end of the Iteration. They should be able to talk about past experiments they have run, and have a mechanism to decide about future experiments based on learning from the past, or new ideas that team members have. The improvement goals / experiments should be documented on the Team page so that others can learn as well.

As said, a Retrospective meeting is the easiest way to get this all done, ensures that the team takes time to focus on improvement, and helps with team dynamics by having the team work complex issues. You will want to have a very good reason for not doing a Retrospective meeting before deciding on some other approach. And, if you don't do this through a meeting, you will need to ensure that you take the time to focus on continuous improvement and that the equivalent results are part of your team documentation.

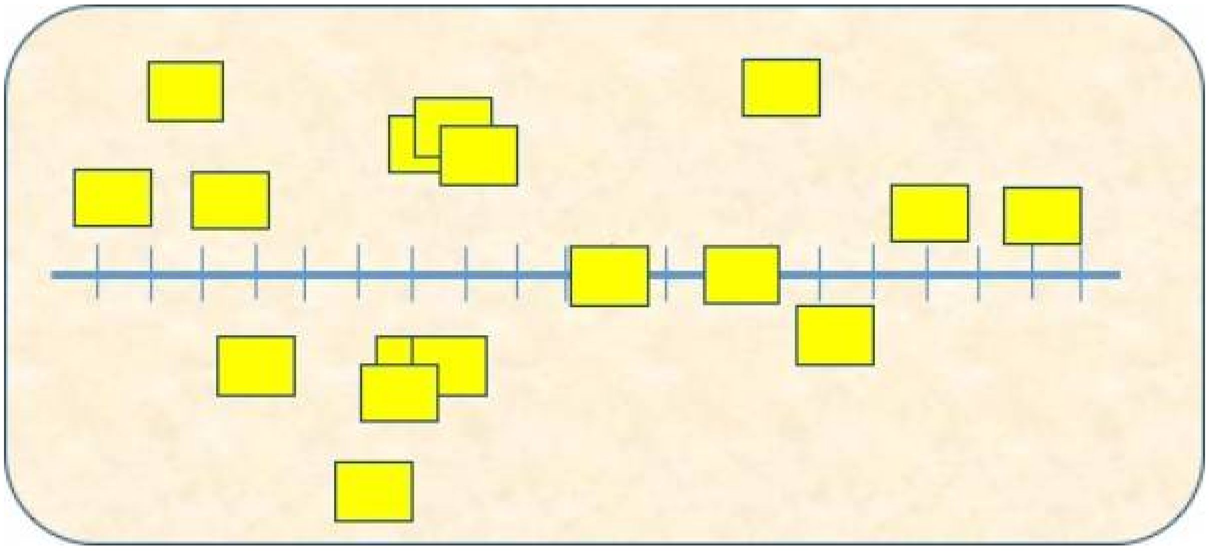
For distributed Teams, there is always an issue of “communication” and so you may want to consider this subject as a standard subject for every Retrospective the Team has or, in some cases, establish a single subject Retrospective to address a specific issue. Many people make the mistake of treating “communication” as a tool issue. In reality it is often an issue of working agreements a Team has (see How Can We Work More Effectively with Remote People? for more information).

"Running" Retrospective

One idea that might help Teams ensure material is available to discuss at the formal Retrospective session or help in those situations where the meeting is not possible is to post a “Running” Retrospective sheet in the Team area (or central tool page if the Team is distributed).

Some ideas for the format of the Running Retrospective:

- Have a series of columns labeled “Keep,” “Change,” and “Add / Learn” in some central place (team room or electronic). Team members would be encouraged to write ideas they have during the course of the Iteration (e.g., on a Post-It© note) and place them under the appropriate column. Then, during the formal Retrospective session, these items can be used to start the discussion.
- More simply the columns might be “Who”, “What Made Me Document It”, “How I Felt About It”, “Ideas to Change It”.
- Another approach is to use a “timeline” where the sprint days are on the X-axis and the Y-axis represents how positive or negative I feel about the item:



Using Metrics

Agile is intended to be an empirical approach to improvement, using data to help drive the process. Example metrics we start with are trends associated with the team's "actual velocity", "committed vs actual velocity" and "unplanned (interrupted) time". These should be used to help understand whether things are improving.

Less formally, as teams come up with experiments they want to run, they should also determine what metrics they are going to track to understand whether progress is being made against the intended expectation. Once the metric has served it purpose, stop generating the metric. Don't create "vanity" metrics ("see we are still doing this great") or useless metrics (pages of data which are generated but from which no conclusion / decision is made).

Tears of Joy

One of the problems of Retrospectives is feeling like there is actual improvements being made. An idea to address this is to run a "Tears of Joy" exercise before the retrospective, perhaps as a check-in exercise. This is a simple exercise. You start be saying "before we get started, I'd like to remind ourselves about the progress we have made, where we came from and where we are today. With this in mind please think back over the last couple of Sprints (or whatever time period makes sense). At some time someone did something, or said something that brought 'a tear of joy' to your eyes. What was that event?" Give the folks a couple of minutes and then go around the room. You will be amazed at what you hear and, more importantly, you'll be able to leverage the positive discussion into the rest of the meeting.

I am indebted to Bob Galen for this idea. Note I find this is useful exercise to do in a variety of situations especially when you know there are significant problems to be addressed. By starting off on a positive note, with a reminder that we have successfully worked issues in the past, the current problems seem to become more solvable.

Challenges of a Retrospective

- Having a Retrospective: After a while team's begin to feel they have all the improvements under control and so think they don't need a retrospective. After all, they are perfect. To be clear – there is always room to get better, and so always a need for something like a retrospective.
- All the issues are not the team's fault: For these teams, recommend that the team's maintain two lists: 1) things that the team needs to address 2) things that the organization needs to address. Set the expectation that both lists will be populated.
- Organizational issues are being dealt with: Management has responsibility to the system the team is operating in. In the early days of the team, there are typically many things identified that can and should be addressed by the team and you often see rapid improvement. Some things can only be addressed by management, and the worse thing that can happen is that the team repeatedly raises organizational issues, but nothing is done about it. Management needs to be open and transparent about working organizational issues and become very responsive to these kinds of requests.
- The team does not talk about people issues: Some people think that high performing teams are smooth running teams. The opposite is true. A significant hallmark of high performing teams is "constructive dissent" where there is passionate disagreements about the things that are important to the goals of the team. To make this kind of happen, team members really need to be able to trust their team mates. Trust only develops when you have dealt (as a team) with all the inter-personal issues you have. Trust does not develop where everyone is behaving politely and where people have not worked through the difficult personal issues.
- The team does not see improvement from one sprint to the next: Make sure the first part of the retrospective is focused on "what happened as a result of the last set of experiments we have run – did things get better and, if not, what have we learned." See also the "Tears of Joy" idea above.
- The retrospective is dominated by the same people all the time: There are a lot of approached aimed at getting everyone to participate. The Scrum Master (as facilitator) should worked to make sure everyone's input is heard.
- The meeting is boring: Especially after the 10th retrospective – same activities, often the same results … Switch things up by:
  - Having a special guest Scrum Master come in to facilitate the retrospective
  - Set up a rotating retrospective amongst team members
  - Use resources like tastycupcakes.org to get new and interesting ideas.
  - And so on.

Want to Know More?

- Agile Retrospectives - the Bible!
- Keeping Retrospectives Fresh - report of a webinar. Idea is to vary goal, exercises and environment to keep retrospective engaging
- Retrospective Wiki - resource for sharing retrospective plans, tips & tricks, tools and ideas to help us get the most out of our retrospectives

- A retrospective toolbox from Ben Linders - Exercises to keep things interesting
- "Dial-a-retrospective" or "Ret-ro-mat" - Uses the structure defined by Agile Retrospectives, and a suite of games / exercises to dial up a new retrospective.
- Tasty Cupcakes Tools for Innovation and Learning - Not just retrospectives, but exercises and games for all kinds of situations.
- Fun Retrospectives

Consultant, Tools, Team, Retrospective, Ceremony, FirstSprint, FAQ