The Scrum Skills Series

Retrospectives, Part I – Getting Started

By

Henrik Berglund, Agile Coach Cedur AB

www.cedur.se

v1.1

Abstract

Running an effective retrospective meeting in a short time is very challenging.

This paper presents a framework for running Scrum retrospectives that I and others have used a lot and that have given us good results. It covers how to prepare for the meeting as well as the purpose and mechanics of each step. It also contains a lot of notes and tips for the ScrumMaster, who will usually be the one facilitating the meeting.

If you would like to improve your retrospectives, I invite you to try this approach out for a few retrospectives. After that, evaluate how it works for you. No matter what, I hope you will have found some new ideas and techniques that you can keep using to improve your future retrospectives even more.

Enjoy the paper!

Contents

Why bother?

Perhaps you are wondering about why you should bother doing retrospectives? How about this:

Retrospectives will make your life better!

Pretty good reason, isn't it!

In a study (1) from 2010, Amabile and Kramer asked hundreds of knowledge workers to write diaries and rate each day as good or bad. After analyzing thousands of entries, they found that "making progress" was linked to 76 percent of employees' reported "best days." Also, "support for making progress" were their No. 1 motivator. "On days when workers have the sense they're making headway in their jobs...their emotions are most positive and their drive to succeed is at its peak."

However, "on days when they feel they are spinning their wheels or encountering roadblocks to meaningful accomplishment, their moods and motivation are lowest."

Making progress and removing roadblocks is what retrospectives are all about, so this is a really good reason to hold them. **When you get your retrospectives working,[1] they will make your time spent at work more enjoyable!**

In this paper I will present some techniques for retrospectives that I have found to work well in practice. My hope is that this will inspire you to run even better retrospectives that will improve life at work for your team!

---

[1] In practice, it is not uncommon that teams struggle a bit with the retrospective and eventually stop holding them to focus on things that they feel actually add value. That's why I wrote this paper. I hope some teams will give the ideas presented here a try and that it will contribute to making their lives at work better.

A framework to use

You usually only have a very short time to run a retrospective meeting, typically 1-2 hours. The structured approach I will show you below has many advantages when it comes to getting the most out of that time. So here we go:

Step1 – Check in

The longer it takes before a person speaks at a meeting, the greater the risk that he/she starts dozing off and starts thinking about other things.

So, if you are the meeting facilitator, don't talk and talk at the start of the meeting!

Instead get everyone involved from the start by asking a simple question. E.g. "Summarize how you feel about the last sprint in <u>a word or two</u>". Make this a quick pass around the table. Get it off to a good start by setting an example yourself, e.g. say "frustrated, hopeful"

Step 2 – Timeline
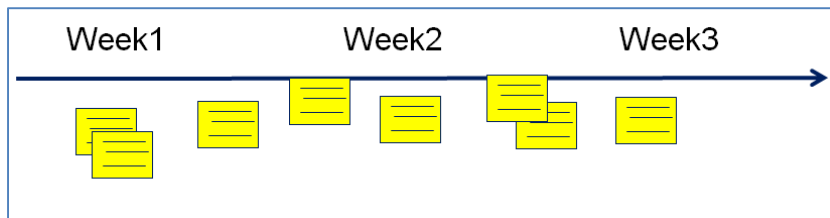
Draw a line on a whiteboard, mark the weeks of your sprint on the line. Ask everyone to write down events that happened during the sprint that were of importance to them on sticky notes. This can be small and big things, good, bad and whatever.

Let people first write for a few minutes, then facilitate that one person at the time post their notes on the timeline. Make sure that as they post them, they also read it out/explain it to the others. If the rest of the team has questions here about what the person means, it is ok to ask to clarify. Besides from that, only the person that hangs up the note talks.

One purpose of this exercise is to jog the memory of the team.

Also, even if you have worked together in a team room during the complete sprint, everyone will not have the exact same perception on why things played out the way they did. By creating the timeline we build a common history of the events that has passed.

Why should only one person at a time talk? Because this makes the process move fast and it gives everyone a chance to make their voice heard without interruption. Also, it minimizes risk for conflict and avoids filling the time with lengthy discussions about non-prioritized issues.

Step 3 – What was good?
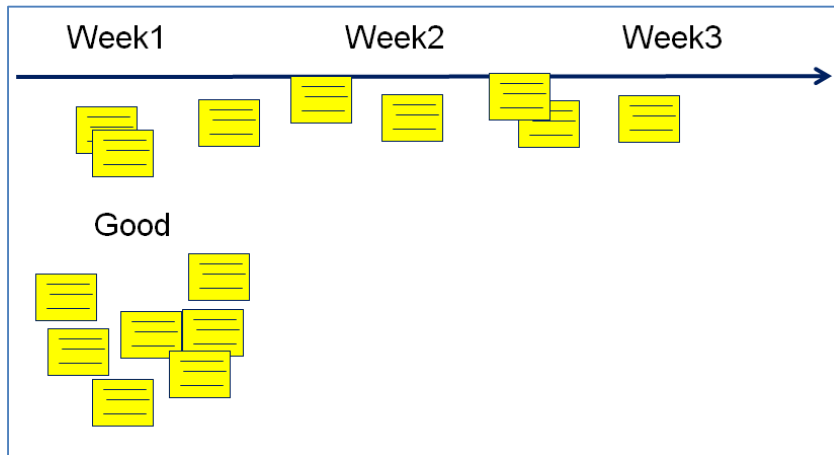
Next, let everyone write sticky notes about all good things that have happened during the sprint. The mechanics are the same as in the previous step. One person at a time posts and at the same time the person reads it out/explains it. Except for questions clarifying the intention of the note, nobody speaks but the person that wrote it. It is ok to move your own notes from the timeline also.

One purpose of this step is to take the time to acknowledge all the positive things that have happened. If you only focus on problems in your retrospectives, it won't be a very fun meeting. After a while you might not really look forward to another full hour of complaining. Also, these can be great sources of ideas on what to do even more of during the next sprint.

Another reason is that we want to take our time to build a safe and positive environment before moving on to the next step where we will be dealing with things that we would like to improve on. These can be sensitive issues. If we can establish a safe and positive feeling before we start with that, more things will get out in the open.

Perhaps on a slightly unrelated note: Researchers have found that a lot of the conflict that goes on in marriage is due to personal differences that are not possible to resolve. Instead what makes marriage work is not the absence of conflict. It is the ratio of positive/negative comments that matters. At least a five to one ratio is needed to make the relationship last!

I read somewhere that for teamwork the corresponding ratio is three to one! So, don't be shy when it comes to listing all the good things that has happened! A lot of positive events are needed to balance the everyday troubles that all teams are facing.

Step 4 – What can be improved
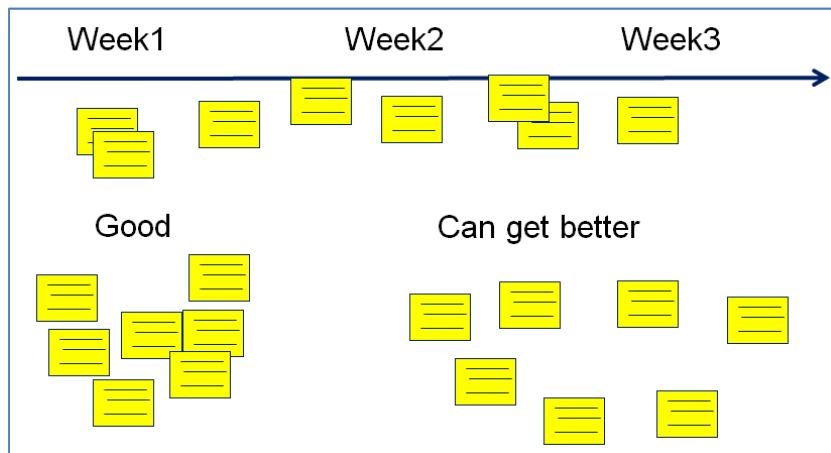
In this step each person writes notes about things that could be improved. Except for that, the procedure is exactly the same as in the previous two steps.

Ask people to just write down what they perceived were problems and to avoid writing down suggested solutions or analysis of problems on the notes.

Team will discuss probable causes and possible solutions later. If someone posts a note with an analysis or a solution proposal, ask them about the perceived problem and ask them to change the note to reflect that.

There is a human tendency that affects retrospectives a lot. It has to do with how the brain handles problems. The default behavior of humans all over the world when something bad happens is to first try to blame the problem on somebody(!) If we move on from that, we next try to blame it on the circumstances. If we move on from that, next we blame it on our own limitations or personal flaws.

While each of these thoughts may be factually correct, the problem with this natural line of thinking is that these are not good problem solving mindsets. It is not likely that your problem will get fixed if you frame it like this in your mind. Watch out for this kind of thinking in the retrospective and act as a facilitator, helping people to write down actual problems first. We will analyze and discuss these later.

This tendency to blame in our mind is described in a simple but brilliant way in the Responsibility Redefined™ (3) process by Christopher Avery. I try to help all teams I meet to learn about this. Teams that do know the process can put a "responsibility poster" on the whiteboard during the retrospective. This will serve as a reminder not to run away from problems by blaming others, blaming the circumstances etc. Instead they can start thinking about what they can do to fix problems they care about.

Step 5 – Group and prioritize

Now you have lots of notes with things that have worked well and things that could be improved on the board.

Get the team to stand in front of the board. Basically we don't want the same thing to be on multiple notes. Ask everyone to work together to group similar notes together and to assign a name to them that summarize the content on all of them.

Then count the number of groups and non-grouped notes and divide this by two or three. This is the number of votes that each team member gets. For example if you have 10 issues, pick a number from 3 to 5.

Each person then casts the selected number of votes on the subject/theme that he/she would be most interested working on during the next sprint. It is no point selecting things that nobody wants to work on. The votes are laid by making one dot per vote using a whiteboard marker. You can put all dots on one group/issue or split them as you like over several issues.

The reason we are including the good things in the vote here is that one of the really best ways of improving is to build on what is already working! Just do more of what works, and try to do that even better. Some teams only do this part and report it works very well!

Count the votes. We now have a prioritized list with subjects that the team would like to work on. Pick one or two that you would like to continue working on.

Step 6 – Analysis (optional)

If time permits, for each issue that you selected, discuss what the root cause(s) might be. A simple way to do this is to use "Five why" analysis. E.g.

- Why did we miss the deadline? -> Development of new features was slow.
- Why was it slow -> Because we were interrupted by lots of trouble reports.
- Why were we interrupted by a lot of trouble reports? -> Because we release a lot of defects to our customers.
- Why did we release a lot of defects to our customers? -> Because we did not test the release properly.
- Why did we not test the release properly? ->Because we did not have time to run the test suites.
- Why did we not have time to run the test suites? ->Because they are all manual and it takes days?
- Why are they all manual? ->Because we don't have time to/don't know how to write automated tests.
- …
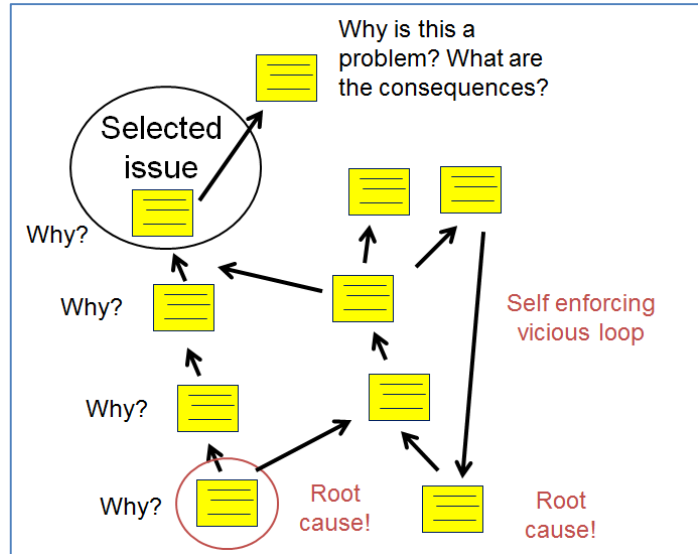
When you have done this for a while (typically at least five times) you will end up with one or several root causes. Addressing these will be more effective than addressing the more obvious symptoms

A way I like even more is cause-effect diagrams, see the example in the illustration above. A great description about that has been made available by Henrik Kniberg (4). Creating cause-effect diagrams takes some time though. If you have a two hour retrospective you usually can make some good progress on it.

If you have shorter retrospectives and/or some difficult problems, consider bringing the cause—effect analysis as an item to your sprint planning and ask the product owner to allocate some time to it during the next sprint.

For working on bigger things a really interesting approach is the A3 process pioneered at Toyota. I have not seen it used much in software development organizations, but I mention it here anyway since I think that it could really be a great way to get tough issues resolved. If you are interested in that, a great book on the subject is "Managing to Learn" by John Shook (5).

Step 7 – Countermeasures

Now list all issues you selected, or their root causes if you took the time to find them in the previous step.

For each issue try to think of a few countermeasures that you think would help mitigate the situation. Next **select one or two** that you would like to try during the next sprint.

Record countermeasures/actions on a flipchart. Then for each countermeasure/action:

- Work with the team to define what you want the situation to be after you have applied it
- Record how you will check if this result have been achieved
- Ask who wants to work on this. Do not assign people to work on things. You get much better results with volunteers. Record the names of persons working on each topic.
- Ask the people involved when it will be finished. Record dates for checking work

Almost done! After the session, bring the flipchart to the teamroom and hang it on the wall.

Step 8 – Plus/Delta

When you are trying out new things, a very good habit is to end each working session with a short plus/delta exercise. On a flipchart/whiteboard, create a table like the one above and ask people what they found valuable about the working session you just had (plus) and what they would like to do differently (delta), if you were to work together like this again.

Let people speak up and record their observations. You will probably get a few ideas on how to make the session better already the next time. Also, you will probably end on a positive note since people speak up and appreciate a lot of things that happened during the session.

This all is useful for the meeting facilitator to be able to help create a great next working session. Talking about how to work together like this, with a very short feedback loop, also has several important team building effects.

Suggested times for each step

Depending on the length of your retrospective, here are some suggestions on how to use the time during the retrospective:

	Retrospective length		
Activity	**2 hours** *(standard length for three week sprints)*	**1,5 hour** *(standard length for two week sprints)*	**1 hour** *(some people prefer this)*
Check in	5 min	5 min	5 min
Time line	15 min	15 min	10 min
What was good?	7 min	7 min	7 min
Can be improved	7 min	7 min	7 min
Group and prioritize	5 min	5 min	5 min
Coffee break	10 min	-	-
Analysis	30 min	10 min	-
Countermeasures	20 min	20 min	10 min
Plus/delta	5 min	5 min	5 min
Switch time (for use to switch between activates)	16 min	16 min	11 min

If you use really short retrospectives, like 1 hour, don't expect to analyze and resolve difficult issues during the retrospective. You can resolve quite a lot though and at least identify and prioritize more difficult problems. . You can then ask for time to keep working on analyzing them in the next sprint planning meeting.

Some teams that do one week sprints hold the retrospective every other week so that they can allocate more time to it.

Props, what you need

- A room, preferably not your team room. The idea with the retrospective is to take a step back and look at the situation from some distance. A room at another site is ideal. For a short retrospective that usually is inconvenient, but at least try to get at room away from the daily action.

- A whiteboard, preferably a big one with lots of space in front of it so that the complete team can sit down in a semi-circle or all stand up and work together.

  Try to not have a table in front of the board. There will be lots of running around it to get to the board. This will just hinder cooperation and participation. Ideally we would have just an empty room.

- A timer to enforce the time-box of each step.
  If the timer goes off and you did not finish the step, move on anyway. This is the purpose of the time-box. Your job is to practice so that you can get the most out of the allocated time. If you are not happy with the results, just accept that this was what you were capable of today. Think about what you could improve in the meeting and try to do better next time.

- Sticky notes and markers for everyone.
  You may want to get the "super sticky" kind depending on the type of board you use. They should be no smaller than 70x120 mm (or text will be too small to read). You should also get proper pens. Permanent markers with medium size tip's work fine. Most other pens that people use for personal notes will not be readable from a distance.

- Flip chart to record countermeasures and plans.
  Just bring the sheet to the team room after the meeting and hang it on the wall.

- A camera (optional)
  In general there is no need to document everything. Soon there will be another retrospective and then the issues that matters most at that time will be raised again. If you want to continue working on a tricky issue later, perhaps using cause-effect analysis and/or the A3 process, a picture of results so far can be useful though.

Facilitation tips

Before the team arrives to the meeting, prepare a list of the steps in the procedure as well as the time-box for each one on a flipchart. Plan the whiteboard space, make sure you have space for the timeline, good and "could be improved" sections.

When introducing this to your team, get their consent before trying the new approach:

- "I would like us to try this process, would you be willing to do that?"

Also ask permission at the beginning of the meeting to facilitate it:

- "I would like us to try to follow the rules to see how it feels like if we do it according to this procedure, so is it ok if I sometimes interrupt you if we start moving outside this format?"

If you get permission by the team, you can start facilitating the meeting.

Don't explain the whole framework in the beginning; just get started as quickly as possible. Half a minute should be enough for you to get permission as explained above. Then explain the purpose of each step as you go along. Don't talk a lot at the beginning of the meeting; people will start to doze off if you do …

When explaining the timeline exercise, lead by example. Write a demonstration note in big readable capital letters, tear it off (sideways without bending it, so that it will stay up better), put it on the wall and read it out.

During timeline, and the steps following that, have some patience and eventually it will get started. If people run out of ideas when there is a lot of time left, prepare some areas of events that you can suggest to try to spark some more memories. Here's a short list from the top of my head:

- Did you make any deliveries? Finish any stories?
- Any nice quality improvements?
- Did you talk to users, what did they say?
- Did you learn something?
- How did the team work together?
- Were you disturbed?
- What other problems were there?

During the early parts of the meeting, people sometimes start talking about causes to problems and solutions to problems. Here are two suggestions on how to handle that:

- "Excuse me! This is a great discussion, but could we postpone it until later in this meeting, when we have all issues on the table and have prioritized what to work on? I just want to make sure we spend our time focusing on the issue we select as most important."

- "Excuse me; I would like us to stick to the rule that only the person at the board talks. Is that ok?"

What if it is not working?

If you are not getting the results you were hoping for you should figure out why before giving up.

Are you sure that you are running the retrospective in the way explained here? If you are not, you should consider practicing until you know how to do it exactly like that. Why? Because until you do, you have not really tried it. You haven't gained any experience on what this particular format can do for you and you have no basis for evaluating it.

When you are sure you are doing it like intended, you can go ahead and evaluate it, compare it with other approaches and adapt it as you feel needed. You need to master it first though.

If you still are having problems, look at what you are doing. What are the problems that you perceive? What could be the root cause of these problems? What could be some things to try to sort out those problems?

Credits

The contents of this paper are based on ideas from many sources, some have been mentioned above. I probably don't remember where I picked up everything, but here are two sources that I have not mentioned so far. First, I learned a lot about the purpose of each step from a talk that Boris Gloger made at QCon London 2007, you can watch the video online (6). I'm sure some of it comes from the standard book on retrospectives, "Agile Retrospectives: Making Good Teams Great", by Esther Derby and Diana Larsen (7).

Most of all, Credit goes to all the teams that allowed me to work with them during their retrospectives to try these ideas out.

Questions, feedback

If you have any questions about this paper, feel free to contact me at henrik@cedur.se. I will do my best to answer promptly.

If people find this first paper in the "Scrum Skills Series" useful, I plan to write similar papers for a variety of topics that has to do with starting up an agile team and keep it evolving.

So if you found the paper useful, or/and have suggestions for improvements, please let me know also!

Future releases in this series will be announced in my blog at www.cedur.se/agile-blog.

Thank you for reading this far and good luck with your retrospectives!

References

1. *What Really Motivates Workes.* **Kramer, Teresa Amabile and Steven.** Harvard Business Review. http://hbr.org/2010/01/the-hbr-list-breakthrough-ideas-for-2010/ar/1.

2. **Kerth, Norman L.** *Project Retrospectives: A Handbook for Team Reviews.* 2001.

3. **Avery, Christopher.** http://www.christopheravery.com. [Online]

4. **Kniberg, Henrik.** Cause-effect diagrams. *www.crisp.se/henrik.kniberg/cause-effect-diagrams.pdf.* [Online]

5. **Shook, John.** *Managing to Learn: Using the A3 management process.* s.l. : Lean Enterprise Institute, Inc, 2008.

6. **Gloger, Boris.** http://www.infoq.com/presentations/Heartbeat-Retrospectives-Boris-Gloger. [Online]

7. **Larsen, Esther Derby and Diana.** *Agile Retrospectives: Making Good Teams Great.* 2006.

License

Revision History

2010-09-16   v1.0

2012-12-05   V1.1

Looked the paper over and made some improvements. Many of these were suggested by Ilja Preuss in his initial feedback a long long time ago... Some of them are my own.

Step2: New recommendation, let people write first, then post. May get generate a more diverse set of ideas

Step 5: Vote also on "good" things - positive change psychology.

Step 7: Ask for volonteers

Step 8. Added plus/delta exercize

Props: Added recommendation of putting chairs in semi-circle in front of board.