



Commit

Holding a Non-Traditional Retrospective

Mary Thengvall, Director, Developer Relations, Camunda



We're going to cover...

- What is a retrospective?
- How should a community retrospective be conducted?
- When the retrospective is over, what happens next?



What is a retrospective? 🤔



What is a retrospective?

A Retrospective is a ceremony held at the end of each iteration in an agile project. The general purpose is to allow the team, as a group, to evaluate its past working cycle. In addition, it's an important moment to gather feedback on what went well and what did not.

5 Things You Need To Know To Facilitate A Retrospective, José Alberto Zimermann



What is a retrospective?

... the open source community version

- Public, not private



What is a retrospective?

... the open source community version

To the company, I represent the community.

To the community, I represent the company.

I must have both of their interests in mind at all times.



What is a retrospective?

... the open source community version

- Public, not private
- Follow-through is (even more) important



What is a retrospective?

... the open source community version

- Public, not private
- Follow-through is (even more) important
- Code of Conduct must be enforced



But why? 🙄

- Encourages open and honest feedback



But why? 🙄

- Encourages open and honest feedback
- Gives you a tangible way to keep track of feedback



How should a community retrospective be conducted? 



Prep: Questions to Ask

- Who do you invite?



Prep: Questions to Ask

- Who do you invite?
- What expectations do you have?



Prep: Questions to Ask

- Who do you invite?
- What expectations do you have?
- Who is officially responsible to run the retrospective?



Prep: Questions to Ask

- Who do you invite?
- What expectations do you have?
- Who is officially responsible to run the retrospective?
- Do you have the bandwidth to follow up on action items?









Execution: Setting Expectations

- Establish working agreement



Execution: Setting Expectations

> Working Agreement    5  0  

Radical Candor
#4

Try to minimize distractions
#5

Don't interrupt others
#6

**When criticizing or saying something negative,
focus on actions, not people**
#7

Timeline
#8



Execution: Setting Expectations

- Establish working agreement
- Identify roles



Execution: Setting Expectations

- Establish working agreement
- Identify roles
- Confirm timeline



Execution: Setting Expectations

Timeline:

- 0:00-5:00: review previous action items if necessary; add last-minute cards. (5 minutes)
- 5:00-8:00: read What's Going Well (3 minutes)
- 8:00-10:00: vote on What's Going Well (2 minutes) - remind folks they get two votes, and they also have a wildcard that they can put in either column
- 11:00-25:00: discuss top What's Going Well and create action items, if any (14 minutes)
- 25:00-28:00: read What's Not Going Well (3 minutes) - remind folks they get two votes, and if they haven't used their wildcard yet, they have a third available
- 28:00-30:00: vote on What's Not Going Well (2 minutes)
- 31:00-50:00: discuss top What's Not Going Well and create action items, if any (19 minutes)
- 50:00-end: finalize and assign action items, meta-feedback (10ish minutes)

It's ok if things move a little faster than this, but let a good long moment of weird silence elapse before deciding to skip forward. Leave at least five minutes at the end to parse and assign action items.



Execution: Walking Through Issues

> **What's Going Well?** 🥳

🗑️ 📄 2 👤 0 + ⚙️

My feedback was accepted!! 🎉

#9

New Community Manager is kicking butt!

#13

> **What's Going Poorly?** 😞

🗑️ 📄 3 👤 0 + ⚙️

Increase Diversity in Community

#10

The new onboarding tutorial has errors

#11

Gaps in the Getting Starts docs

#12



**Once the retrospective is over,
what happens next?**



Following up on Action Items

- Communicate feedback
- Update documentation
- Create content
- Schedule meeting



Following up on Action Items

The screenshot shows the GitLab web interface for an Issue Board. The top navigation bar includes the GitLab logo, 'Projects', 'Groups', and 'More' menus, along with a search bar. The left sidebar contains a navigation menu with options like 'Open Source Retrospective Template', 'Project overview', 'Repository', 'Issues' (with a count of 10), 'Boards', 'Merge Requests' (with a count of 0), 'Requirements', 'CI / CD', 'Security & Compliance', 'Operations', 'Packages & Registries', 'Analytics', 'Wiki', 'Snippets', 'Members', and 'Settings'. The main content area is titled 'Mary Thengvall > Open Source Retrospective Template > Issue Boards'. It features a search bar and a dropdown menu set to 'Action Items'. Below this, three boards are visible: 'Action Items' (with a green checkmark icon), 'Give Feedback' (with a green checkmark icon), and 'Schedule Meeting' (with a red checkmark icon). Each board has a header bar with icons for trash, copy, share, and settings.



Summary

- **Public retrospectives give you insight into your community.**
- **Transparency is key.** When is the retro taking place? How can people take part? What happens after the retro? What's the status of the items that were discussed in the last retro? Communicate early and often.
- **Set expectations.** What do you expect from community members? What can they, in turn, expect from you?
- **Follow through.** Don't make promises you can't keep.



Thank You

Mary Thengvall
Director, Developer Relations, Camunda
@mary_grace
marygrace.community

https://gitlab.com/mary_grace/open-source-retrospective-template