

The background of the slide is a grayscale aerial photograph of the University of Porto's campus, showing a dense cluster of modern and traditional university buildings.

M.EIC, 2022-23

Large Scale Software Development

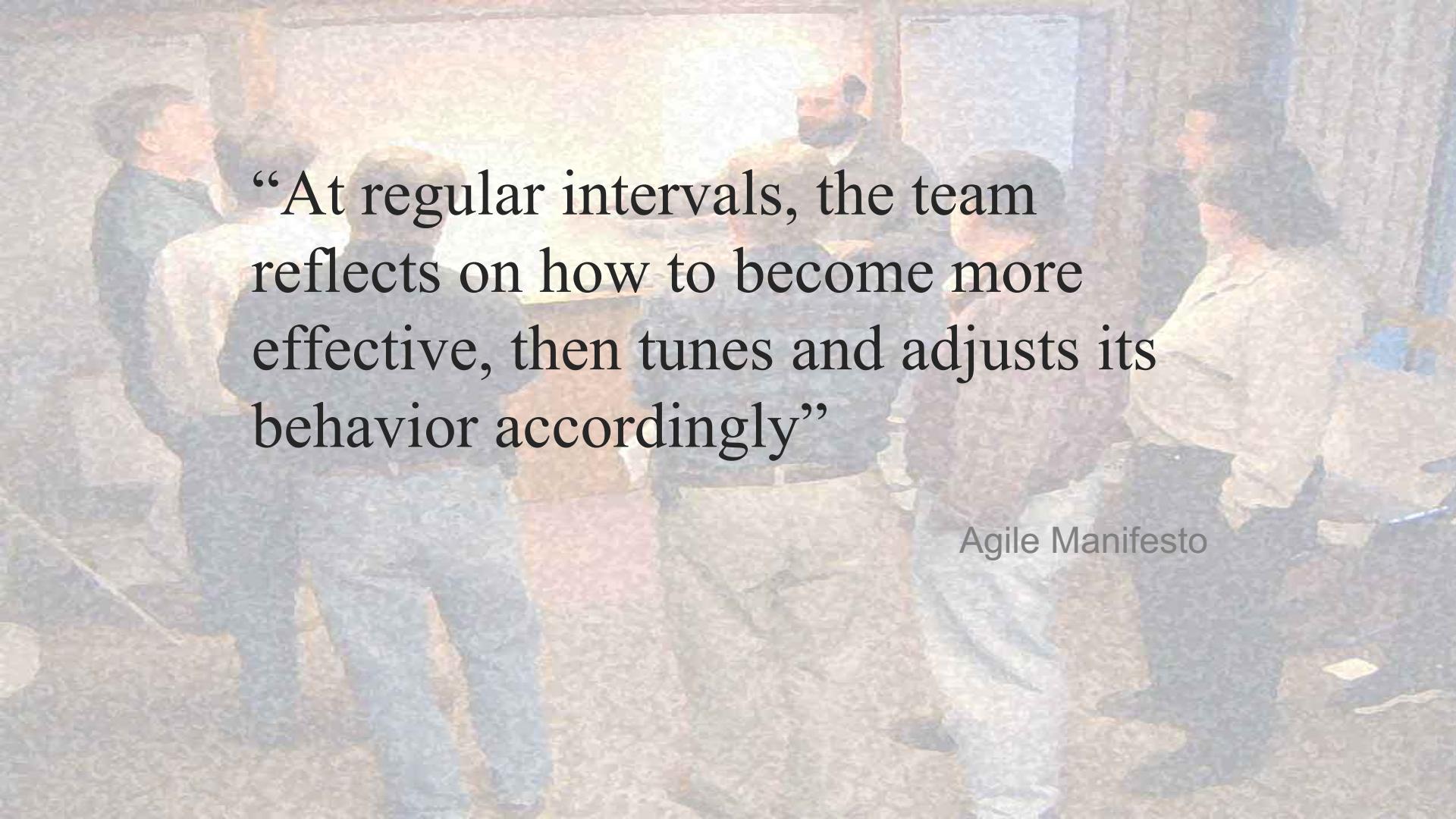
Filipe Correia, Daniel Pinho, João Pedro Dias



How do teams learn and improve?

Inspect and Adapt!

- A big part of agile is about feedback.
- How is our code doing? (e.g., automated tests, code reviews, etc.)
- Retrospectives are a way we inspect and adapt our methods, engineering practices and teamwork.

A photograph of a group of people in a workshop or laboratory setting. They are wearing lab coats and safety glasses, and are focused on a task at a table covered with various pieces of equipment and materials. The scene is slightly blurred, creating a sense of activity and concentration.

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

Agile Manifesto

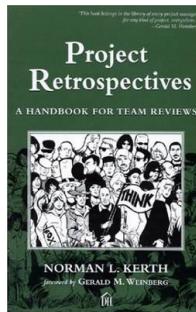
“We want to believe that learning from experience is automatic, but it requires profound skills. Experience provides data, not knowledge.”

The Power of Retrospectives – Linda Rising

What are Agile Retrospectives?

“A ritual held at the end of a project
to learn from the experience and to
plan changes for the next effort.”

Norman Kerth
retrospectives.com

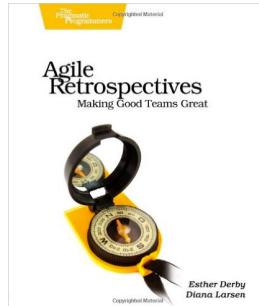


“Agile retrospectives are special meetings that take place at the end of a period of work, usually an iteration or release.”

Esther Derby

“In a retrospective, a team steps back, examines the way they work, analyzes, and identifies ways they can improve.”

Esther Derby

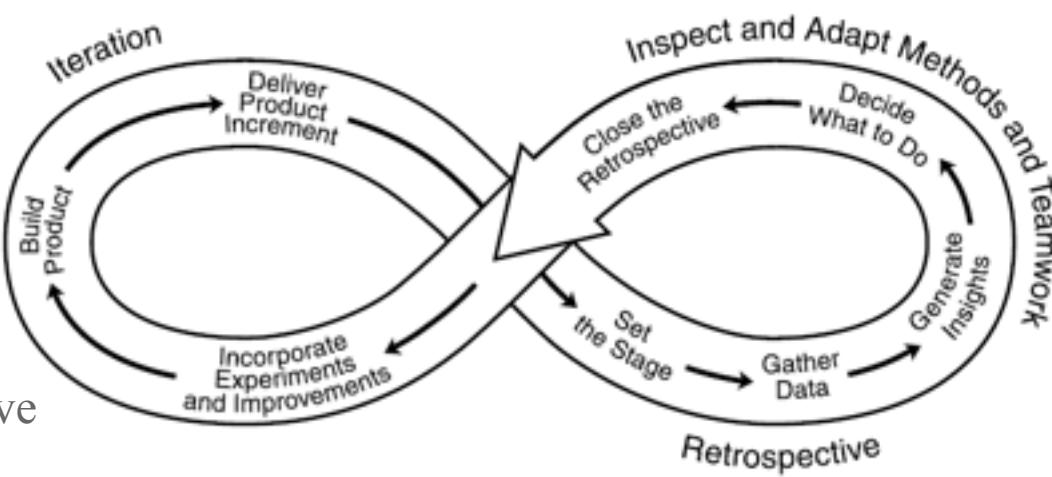


Prime Directive

“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.”

Steps of a Retrospective

1. Set the Stage
2. Gather Data
3. Generate Insights
4. Decide What to Do
5. Close the Retrospective



① Set the Stage

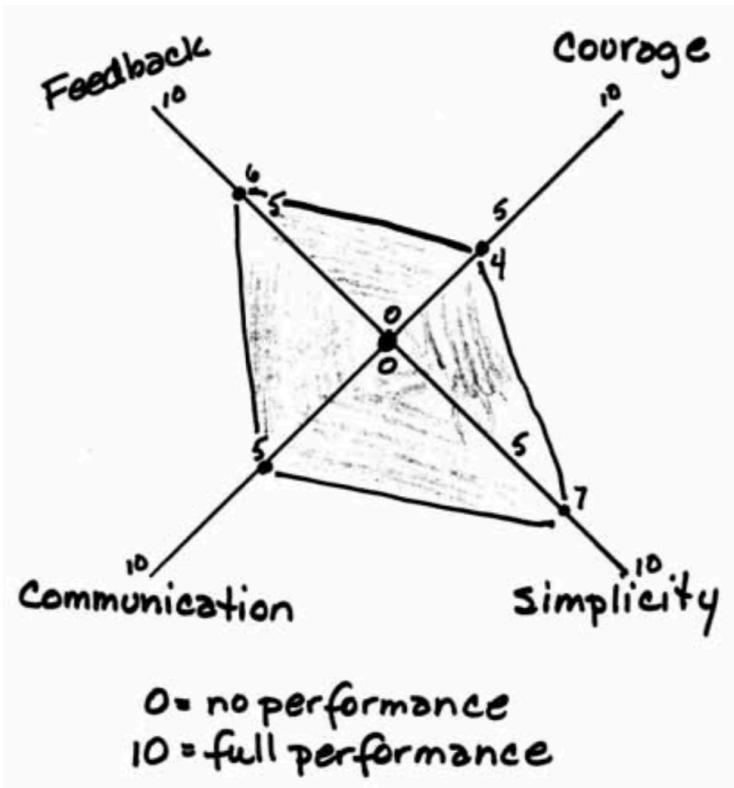
- Establish the timebox.
- Establish the goal for this retrospective.
 - General goal: “continuous improvement”
 - Example of focused goal: “are we doing Pair-Programming right?”
- Establish the rules.
 - Agree on how we're going to work together
 - Includes making sure everyone is aware of the Prime Directive

②

Gather Data

- Look back to the iteration and collect data.
- Helps the team “think together”.
- Get a shared picture of events.
- Different sources and methods:
 - Timeline (facts)
 - Emotional seismograph (people)
 - Version-control data, customer feedback and other time-bound project information may also be useful.

Team Radar



Emotional Seismograph



<http://www.flickr.com/photos/garrettc/2575214144/>

③

Generate Insights

- Asking “why?”
- Data supports a shared understanding
- Several methods ...

- > Example: Learning Matrix

- What went well?

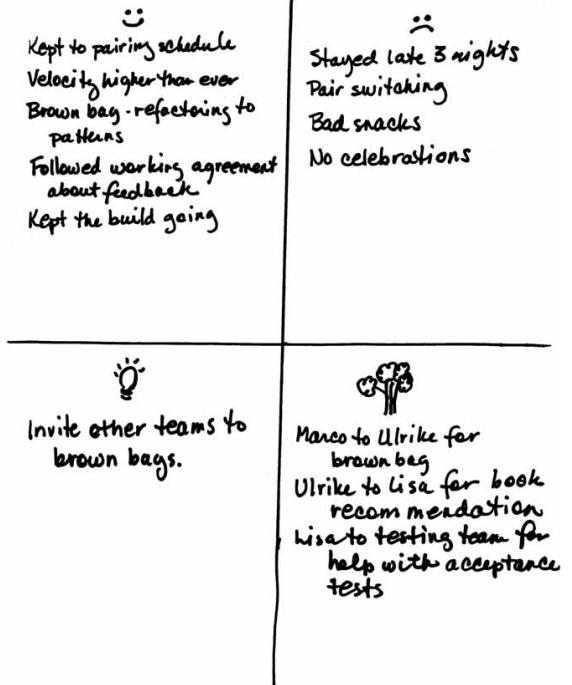
- What we want to do differently?

- Ideas

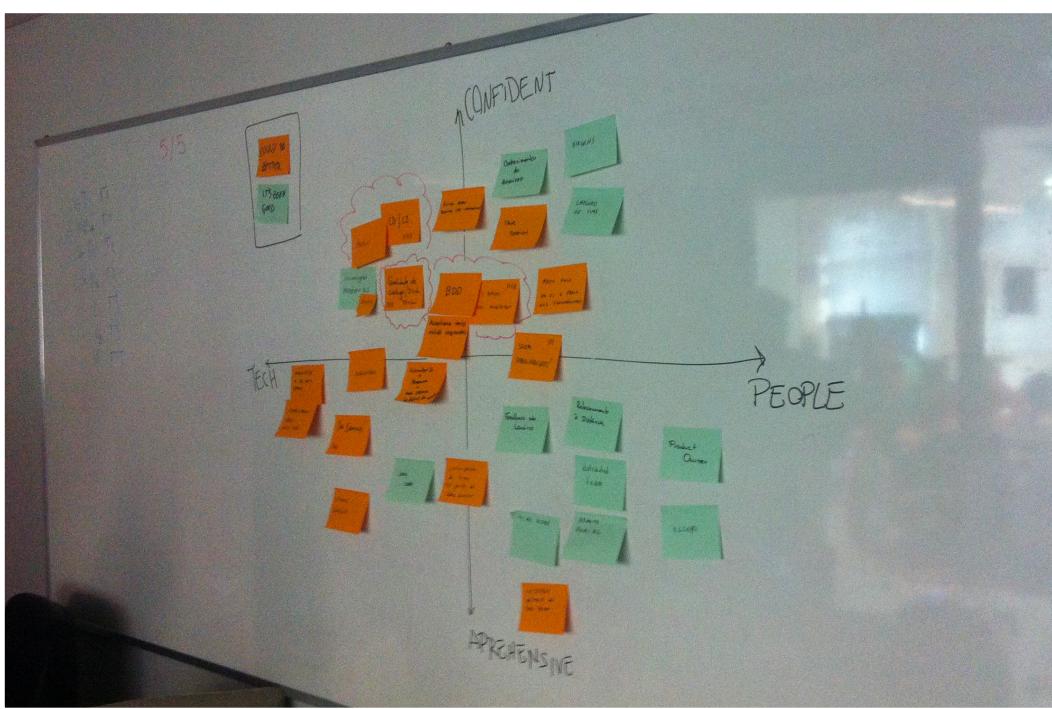
- Appreciations

- > Many others ...

- Five whys, CAPT, Brainstorming/filtering, etc.



Confident or Apprehensive × People or Tech



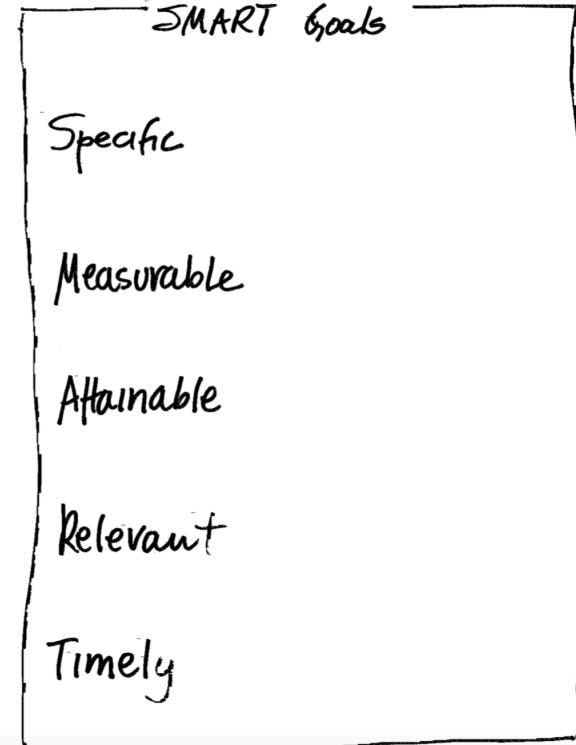
④

Decide What to Do

What can we do during the next iteration towards our goal?

- Generate several ideas ...
- Pick only a couple of things to try
- If there are more ideas than the team can chew, prioritize. Give higher priorities to:
 - > the ones to which the team is **most motivated**
 - > the ones with the **most impact**
 - > the ones **easier** to take on

SMART Goals



⑤

Close The Retrospective

- End the retrospective decisively
- Recap on what was decided
- Decide on how the follow up will be done
- Ask how can the retrospective itself can be improved

What not to expect from a retrospective

- It's not the place for individual performance feedback.
- Can't fix inadequate technical or collaboration skills.
- Can't fix faulty mental models of software development or how organizations work.

Pitfalls

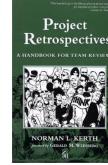
They can become boring, painful or a waste of time. Why?

- > **Lack of preparation** – There should be an agenda.
- > **Lack of focus** – Focus on a specific topic. “Continuous improvement” is (only) a good topic to start with.
- > **Lack of data** – Talk about what happened first, then talk about what to change.
- > **Lack of balanced participation** – Everyone should have a chance to participate and contribute to the result. Use small-group discussions to ensure it.
- > **Lack of empowerment** – Focus on choosing improvements the team can do themselves or on which they have strong influence.
- > **Lack of energy** – The team should have enough motivation and velocity to realistically incorporate the actions in the next iteration.
- > **Lack of incorporation into the day to day work plan** – Write a user story card for the chosen improvement, and take it into the next iteration planning meeting.

References

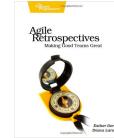
Project Retrospectives, by Norman Kerth

Book – <http://www.amazon.com/Project-Retrospectives-Handbook-Team-Reviews/dp/0932633447>
Website – <http://retrospectives.com>



Agile Retrospectives, by Esther Derby & Diana Larsen

Book – <http://www.amazon.com/Agile-Retrospectives-Making-Teams-Great/dp/0977616649>
Presentation – <http://www.slideshare.net/estherderby/agile-retrospectives-4976896>
Google Tech Talk – <http://www.youtube.com/watch?v=qqtPZYigfNI>



Live and Learn with Retrospectives, by Rachel Davies

<http://www.infoq.com/articles/agile-retrospectives-davies>

agileretrospectivewiki.org

The Power of Retrospectives, by Linda Rising

https://gotocon.com/dl/jaoo-sydney-2009/slides/LindaRising_ThePowerOfRetrospectives.pdf

FEUP.DEI



DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
dei.fe.up.pt | secdei@fe.up.pt | +351 225 082 134
Director **João MP Cardoso** | Full Professor
jmpc@fe.up.pt | +351 220 413 284

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Rua Dr. Roberto Frias | 4200-465 Porto | PORTUGAL
www.fe.up.pt | feup@fe.up.pt | +351 225 081 400
 @DEI_FEUP