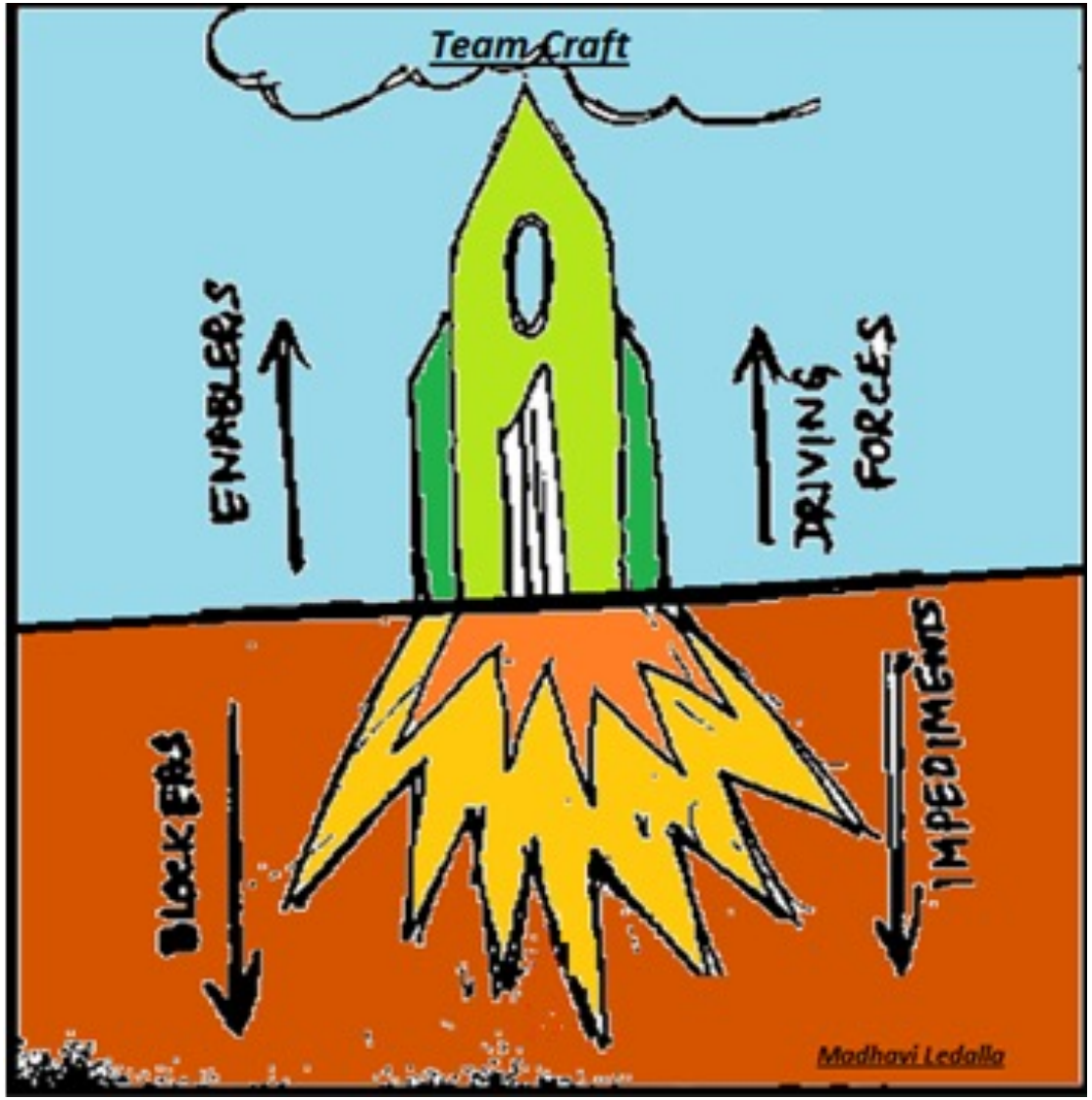
Sprint Retrospective



From *7 Ways to Make Retrospectives Fun and Engaging* by Vibhu Srinvasan

SWEN-261
Introduction to Software Engineering

Department of Software Engineering
Rochester Institute of Technology

Software Engineering
Rochester Institute of Technology

Team retrospectives are all about improving the team and your process.

- The team reflects (introspects) on three main questions:
  - ***What went well?***
  - ***What didn't go well?***
  - ***What can we do to improve?***
- Do a retrospective on each sprint.
  - ***This allows a team to make frequent*** *course corrections* ***(aka improvements)***
  - *Iterative and incremental* ***is a principle to be applied to your process (as well as the product)***
- There are dozens of specific retrospective techniques; we'll teach you one.

The starfish technique uses five categories of issues.

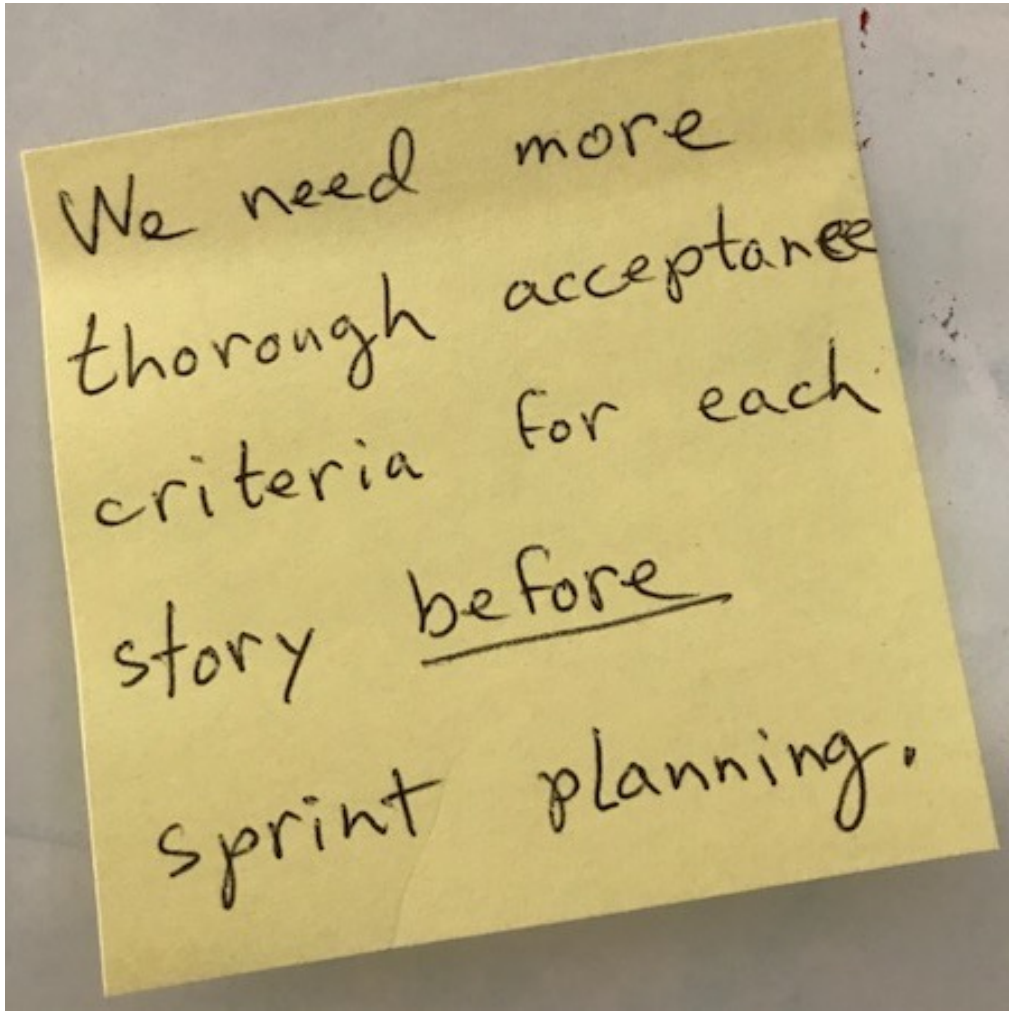
- Keep doing
  - ***These issues highlight an activity that worked well***
  - ***No change necessary***
- More of
  - ***These issues request more of an activity***
- Start doing
  - ***These issues request the start of a new activity***
- Less of
  - ***These issues request less of an activity***
- Stop doing
  - ***These issues request stopping an activity that isn't serving the team, the product or the stakeholders***

The process of a retrospective follows these steps.

1. Every member creates issue cards
2. Members place each card on the starfish chart
3. A facilitator reads aloud each issue and groups common issues together
4. Every member votes for five issues on the chart
5. The facilitator picks top three issues
6. The team brainstorms on solutions to each of these top issues
7. The team creates action items for the next sprint to satisfy the top issues

Keep your issues concise yet complete.

- Here's an example:



Issues run the gamut from process, design, teamwork and communication.

- Adding, removing or improving processes:
  - ***backlog refinement***
  - ***sprint planning***
  - ***calculating team capacity (velocity)***
  - ***daily standups***
  - ***sprint review/demo***
  - ***sprint retrospective (the meta process)***
- Adding, removing or improving reviews:
  - ***code reviews***
  - ***reviewer engagement***
  - ***pre-development design reviews***
  - ***pre-planning design reviews***

Issues run the gamut from process, design, teamwork and communication.

- Adding, removing or improving testing:
  - ***unit testing***
  - ***unit testing code coverage***
  - ***integration testing (esp for web apps)***
  - ***acceptance (manual) testing***
- Adding or improving developer or team skills:
  - ***better use of Spikes***
  - ***addition of tech talks***
  - ***formal member training***
- Adding or improving team communication:
  - ***more or better use of communication tools***
  - ***more or better face-to-face meetings***
  - ***more or better discussions with the Product Owner***
  - ***better virtual (distributed) teaming practices***