

Agile Ceremonies

Sprint Planning

Goals

- The planning supports Diversity and Inclusion principles and provides equal opportunities.
- The Planning defines how the work is going to be completed in the sprint.
- Stories fit in a sprint and are [designed](#) and [ready](#) before the planning.

Note: Self assignment by team members can give a feeling of fairness in how work is split in the team. Sometime, this ends up not being the case as it can give an advantage to the loudest or more experienced voices in the team. Individuals also tend to stay in their comfort zone, which might not be the right approach for their own growth.*

Sprint Goal

Consider defining a sprint goal, or [list of goals](#) for each sprint. Effective sprint goals are a concise bullet point list of items. A Sprint goal can be created first and used as an input to choose the Stories for the sprint. A sprint goal could also be created from the [list of stories](#) that were picked for the Sprint.

The sprint goal can be used:

- At the end of each stand up meeting, to remember the north star for the Sprint and help everyone taking a step back
- During the sprint review ("was the goal achieved?", "If not, why?")

Note: A simple way to define a sprint goal, is to create a User Story in each sprint backlog and name it "Sprint XX goal". You can add the bullet points in the description.*

Stories

Example 1: Preparing in advance

- The dev lead and product owner plan time to prepare the sprint backlog ahead of sprint planning.
- The dev lead uses their experience (past and on the current project) and the estimation made for these stories to gauge how many should be in the sprint.
- The dev lead asks the entire team to look at the tentative sprint backlog in advance of the sprint planning.
- The dev lead assigns stories to specific developers after confirming with them that it makes sense
- During the sprint planning meeting, the team reviews the sprint goal and the stories. Everyone confirm they understand the plan and feel it's reasonable.

Example 2: Building during the planning meeting

- The product owner ensures that the highest priority items of the product backlog is refined and estimated following the team estimation process.
- During the Sprint planning meeting, the product owner describe each stories, one by one, starting by highest priority.
- For each story, the dev lead and the team confirm they understand what needs to be done and add the story to the sprint backlog.
- The team keeps considering more stories up to a point where they agree the sprint backlog is full. This should be informed by the estimation, past developer experience and past experience in this specific project.
- Stories are assigned during the planning meeting:
 - **Option 1:** The dev lead makes suggestion on who could work on each stories. Each engineer agrees or discuss if required.
 - **Option 2:** The team review each story and engineer volunteer select the one they want to be assigned to.

Note: this option might cause issues with the first core expectations. Who gets to work on what? Ultimately, it is the dev lead responsibility to ensure each engineer gets the opportunity to work on what makes sense for their growth.)

Tasks

Examples of approaches for task creation and assignment:

- Stories are split into tasks ahead of time by dev lead and assigned before/during sprint planning to engineers.
- Stories are assigned to more senior engineers who are responsible for splitting into tasks.
- Stories are split into tasks during the Sprint planning meeting by the entire team.

Note: Depending on the seniority of the team, consider splitting into tasks before sprint planning. This can help getting out of sprint planning with all work assigned. It also increase clarity for junior engineers.

Sprint Planning Resources

- [Definition of Ready](#)
- [Sprint Goal Template](#)
- [Planning](#)
- [Refinement](#)
- [User Stories Applied: For Software Development](#)

Estimation

Goals

- Estimation supports the predictability of the team work and delivery.

- Estimation re-enforces the value of accountability to the team.
- The estimation process is improved over time and discussed on a regular basis.
- Estimation is inclusive of the different individuals in the team.

Rough estimation is usually done for a generic SE 2 dev.

Example 1: T-shirt Sizes

- The team use t-shirt sizes (S, M, L, XL) and agrees in advance which size fits a sprint. In this example: S, M fits a sprint, L, XL too big for a sprint and need to be split / refined
- The dev lead with support of the team roughly estimates how much S and M stories can be done in the first sprints
- This rough estimation is refined over time and used to as an input for future sprint planning and to adjust project end date forecasting

Example 2: Single Indicator

- The team uses a single indicator: "does this story fits in one sprint?", if not, the story needs to be split
- The dev lead with support of the team roughly estimates how many stories can be done in the first sprints
- How many stories are done in each sprint on average is used as an input for future sprint planning and as an indicator to adjust project end date forecasting

Example 3: Planning Poker

- The team does planning poker and estimates in story points
- Story points are roughly used to estimate how much can be done in next sprint
- The dev lead and the TPM uses the past sprints and observed velocity to adjust project end date forecasting

Other Considerations

- Estimating stories using story points in smaller project does not always provide the value it would in bigger ones.
- Avoid converting story points or t-shirt sizes to days.

Measure Estimation Accuracy

- Collect data to monitor estimation accuracy and sprint completion over time to drive improvements.
- Use the sprint goal to understand if the estimation was correct. If the sprint goal is met: does anything else matter?

Scrum Practices

While Scrum does not prescribe how to size work, Professional Scrum is biased away from absolute estimation (hours, function points, ideal-days, etc.) and towards relative sizing.

Planning Poker

Planning Poker is a collaborative technique to assign relative size. Developers may choose whatever units they want - story points and t-shirt sizes are examples of units.

'Same-Size' Product Backlog Items (PBIs)

'Same-Size' PBIs is a relative estimation approach that involves breaking items down small enough that they are roughly the same size. Velocity can be understood as a count of PBIs; this is sometimes used by teams doing continuously delivery.

'Right-Size' Product Backlog Items (PBIs)

'Right-Size' PBIs is a relative estimation approach that involves breaking things down small enough to deliver value in a certain time period (i.e. get to Done by the end of a Sprint). This is sometimes associated with teams utilizing flow for forecasting. Teams use historical data to determine if they think they can get the PBI done within the confidence level that their historical data says they typically get a PBI done.

Estimation Resources

- [The Most Important Thing You Are Missing about Estimation](#)

Retrospectives

Goals

- Retrospectives lead to actionable items that help grow the team's engineering practices. These items are in the backlog, assigned, and prioritized to be fixed by a date agreed upon (default being next retrospective).
- Retrospectives are used to ask the hard questions ("we usually don't finish what we plan, let's talk about this") when necessary.

Suggestions

- Consider [other retro formats](#) available outside of Mad Sad Glad.
 - **Gather Data:** Triple Nickels, Timeline, Mad Sad Glad, Team Radar
 - **Generate Insights:** 5 Whys, Fishbone, Patterns and Shifts
- Consider setting a retro focus area.
- Schedule enough time to ensure that you can have the conversation you need to get the correct plan an action and improve how you work.
- Bring in a neutral facilitator for project retros or retros that introspect after a difficult period.

Use the following retrospectives techniques to address specific trends that might be emerging on an engagement

5 Whys

If a team is confronting a problem and is unsure of the exact root cause, the 5 whys exercise taken from the business analysis sector can help get to the bottom of it. For example, if a team cannot get to *Done* each Sprint, that would go at the top of the whiteboard. The team

then asks why that problem exists, writing that answer in the box below. Next, the team asks why again, but this time in response to the *why* they just identified. Continue this process until the team identifies an actual root cause, which usually becomes apparent within five steps.

Processes, Tools, Individuals, Interactions and the Definition of Done

This approach encourages team members to think more broadly. Ask team members to identify what is going well and ideas for improvement within the categories of processes, tools, individuals/interactions, and the Definition of Done. Then, ask team members to vote on which improvement ideas to focus on during the upcoming Sprint.

Focus

This retrospective technique incorporates the concept of visioning. Using this technique, you ask team members where they would like to go? Decide what the team should look like in 4 weeks, and then ask what is holding them back from that and how they can resolve the impediment. If you are focusing on specific improvements, you can use this technique for one or two Retrospectives in a row so that the team can see progress over time.

Retrospective Resources

- [Agile Retrospective: Making Good Teams Great](#)
- [Retrospective](#)

Sprint Demo

Goals

- Each sprint ends with demos that illustrate the sprint goal and how it fits in the engagement goal.

Suggestions

- Consider not pre-recording sprint demos in advance. You can record the demo meeting and archive them.
- A demo does not have to be about running code. It can be showing documentation that was written.

Sprint Demo Resources

- [Sprint Review/Demo](#)

Stand-Up

Goals

- The stand-up is run efficiently.
- The stand-up helps the team understand what was done, what will be done and what are the blockers.
- The stand-up helps the team understand if they will meet the sprint goal or not.

Suggestions

- Keep stand up short and efficient. Table the longer conversations for a parking lot section, or for a conversation that will be planned later.
- Run daily stand ups: 15 minutes of stand up and 15 minutes of parking lot.
- If someone cannot make the stand-up exceptionally: Ask them to do a written stand up in advance.
- Stand ups should include everyone involved in the project, including the customer.
- Projects with widely divergent time zones should be avoided if possible, but if you are on one, you should adapt the standups to meet the needs and time constraints of all team members.

Stand-Up Resources

- [Stand-Up/Daily Scrum](#)
-

Last update: August 26, 2024