# Iteration Planning

*Iteration Planning* is an event where all team members determine how much of the Team Backlog they can commit to delivering during an upcoming Iteration. The team summarizes the work as a set of committed Iteration Goals.

Teams plan by selecting Stories from the team backlog and committing to execute a set of them in the upcoming iteration. The team's backlog has been seeded and partially planned during Program Increment (PI) Planning. In addition, the teams have feedback—not only from their prior iterations but also from the System Demo and other teams they are working with. That, and the natural course of changing fact patterns, provides the broader context for iteration planning. The output of iteration planning is:

- The iteration backlog, consisting of the stories committed to for the iteration, with clearly defined acceptance criteria
- A statement of iteration goals, typically a sentence or two for each one, stating the business objectives of the iteration
- A commitment by the team to the work needed to achieve the goals

# Details

The purpose of iteration planning is to organize the work and define a realistic scope for the iteration. Each Agile Team agrees on a set of stories for the upcoming iteration (the iteration backlog) and summarizes those stories into a set of iteration goals. The iteration backlog and goals are based on the team's capacity and allow for consideration of each story's complexity, size, and dependencies on other stories and other teams. At the end of planning, the teams commit to the goals of the iteration and adjust stories as necessary to achieve the larger purpose. In return, management does not interfere or adjust the scope of the iteration, allowing the team to stay focused.

## Inputs to Iteration Planning

In SAFe, iteration planning is a refinement of the level of detail, and an adjustment of, the initial iteration plans created during PI planning. Teams approach iteration planning with a pre-elaborated Team Backlog. (They have usually held a backlog refinement event during the previous iteration.) There are a number of inputs to the iteration planning event:

- The Team and Program PI Objectives, created during PI planning
- The team's PI plan, which consists of stories that were identified during PI planning and provisionally assigned to iterations
- Additional stories that have arisen from a local context, including items such as defects and Refactors, as well as new stories that have come about since the planning session to support Feature delivery
- Feedback from the prior iteration, including any stories that were not successfully completed (e.g., did not meet the 'definition of done') in that iteration. For more information on the definition of done, please see the 'Scaled Definition of Done' paragraph in the Built-in-Quality article.
- Feedback from the system demo

## Planning the Iteration

Prior to iteration planning, the Product Owner (PO) will have prepared some preliminary iteration goals, based on the team's progress in the Program Increment (PI) so far. Typically, the Product Owner starts the event by reviewing the proposed iteration goals and the higher-priority stories in the team backlog. During iteration planning, the Agile team discusses implementation options, technical issues, Nonfunctional Requirements (NFRs), and dependencies, then plans the iteration. The Product Owner defines the 'what'; the team defines 'how' and 'how much'.

Throughout iteration planning, the team elaborates the acceptance criteria for each story and estimates the effort to complete each one. Based on their available capacity for the iteration, the team then selects the candidate stories. Some teams break each story down into tasks and forecast them in hours to confirm that they have the capacity and skills to complete them. Once this process is complete, the team commits to the work and records the iteration backlog in a visible place, such as a storyboard, Kanban board or Agile project management tooling. Planning is timeboxed to a maximum of four hours for a two-week iteration.

## Establishing Capacity

First, the team quantifies their capacity to perform work in the upcoming iteration. Each team member determines their availability, acknowledging time off and other potential duties. This activity also takes into account other standing commitments—such as maintenance—that is distinct from new story development (see the section about capacity allocation in the Team Backlog article). Using their historical velocity as a starting point, the team makes adjustments based on the unavailable time for each team member to determine the actual capacity for the iteration.

## Story Analysis and Estimating

Once team capacity has been established, the team backlog is reviewed. Each story is discussed, covering relative difficulty, size, complexity, uncertainty, technical challenges, and acceptance criteria. Teams often use Behavior-Driven Development (BDD) to ensure a shared understanding of each story's specific behavior. Finally, the team agrees to a size estimate for the story. There are typically other types of stories on the team backlog as well, including Enablers that could constitute infrastructure work, research Spikes, and architectural improvements, as well as refactoring work and defects. These items are also prioritized and estimated.

## Tasking Stories

Some teams break each story into tasks. As the tasks are identified, team members discuss each one: who would be the best person(s) to accomplish it, approximately how long it will take (typically in hours), and any dependencies it may have on other tasks or stories. Once all this is understood, a team member takes responsibility for a specific task or tasks. As team members commit to tasks, they reduce their individual iteration capacity until it reaches zero. Often, toward the end of the session, some team members will find themselves overcommitted, while others will have some of their capacity still available. This situation leads to a further discussion among team members to evenly distribute the work.

While breaking stories into tasks is fairly common, it is optional and not mandated in SAFe. It is mostly used by new teams to better understand their capacity and capabilities, and over time becomes unnecessary, with the team planning only at the story level.

## Developing Iteration Goals

Once the iteration backlog is understood, the team turns their attention to synthesizing one or more iteration goals that summarize the work the team aims to accomplish in that iteration. They are based on the iteration backlog as well as the team and program PI objectives from the PI planning event. The closer this iteration is to the PI planning session, the more likely the PI objectives will remain unchanged.

## Committing to Iteration Goals

When the team's capacity has been reached, in terms of committed stories, no more stories are pulled from the team backlog. At this point, the Product Owner and team agree on the final list of stories that will be selected, and they revisit and restate the iteration goals. The entire team then commits to the iteration goals, and the scope of the work remains fixed for the duration of the iteration.

# Attendees

Attendees of the iteration planning event include:

- The Product Owner

- The Scrum Master, who acts as the facilitator for this event
- All team members
- Any other stakeholders as required, including representatives from different Agile teams or the ART, and subject matter experts

# Agenda

An example agenda for iteration planning follows:

1. Calculate the available team capacity for the iteration.
2. Discuss each story, elaborate acceptance criteria, and provide estimates using story points.
3. Planning stops once the team runs out of capacity.
4. Determine and agree on the iteration goals.
5. Everyone commits to the goals.

Acceptance criteria are developed through conversation and collaboration with the Product Owner and other stakeholders. Based on the story estimates, the Product Owner may change the ranking of the stories.

# Guidelines

Below are some tips for holding an iteration planning meeting:

- Timebox the event to 4 hours or less
- Iteration planning is organized by the team and is for the team
- A team should avoid committing to work that exceeds its historical velocity

# Relative Estimating, Velocity, Capacity, and Normalizing Story Point Estimating

Agile teams use story points to relatively estimate stories [2, 3]. With relative estimating, the size (expected effort) for each story is compared to other stories. For example, an eight-point story is four times the effort of a two-point story. The team's velocity is equal to the historical average of all the stories completed per iteration. Velocity is the starting point for calculating a team's capacity for a future iteration. Knowing a team's capacity assists with planning and helps limit Work in Process (WIP)—teams don't take on more stories than their prior velocity would suggest is reasonable. Velocity is also used to estimate how long it takes to deliver Features or Epics, which are also forecasted in story points.

## Normalizing Story Point Estimating

In Scrum, each team's story point estimation—and the resultant velocity—is typically a local and independent matter. The fact that a small team might estimate in such a way that their velocity is 50, while another larger team estimates that their velocity is 13, usually does not pose a concern.

In SAFe, however, story point estimation must be normalized, so that estimates for features or epics that require the support of multiple teams, are based on the same story point definition, allowing a shared basis for economic decision-making. One approach is as follows:

- Find a small story that would take about a half-day to develop and a half-day to test and validate, and call it a 'one'
- Estimate every other story relative to that 'one'

## Establish velocity before historical data exists

Additionally, in an initial PI Planning, it is likely that some of the teams will be new and will not have a historical velocity to use in order to predict their capacity for the iteration. The approach below can be applied in these situations:

- For every full-time developer and tester on the team, give the team 8 points (adjust for part-timers)
- Subtract one point for every team member vacation day and holiday in the iteration

**Example**: Assuming a six-person team composed of three developers, two testers, and one PO, with no vacations, then the estimated initial velocity equals 5 × 8 points or 40 points per iteration. (Note: The team may need to adjust a bit lower if one of the developers and testers is also the *Scrum Master*.)

Note: There is no need to recalibrate normalized estimates or initial velocities. It is just a common starting point.

In this way, story points are somewhat comparable across teams. Management can better understand the cost for a story point and more accurately determine the cost of an upcoming feature or epic.

While teams will tend to increase their velocity over time—and that's a good thing— in reality, the number tends to remain stable. A team's velocity is far more affected by changing team size and technical context than by productivity variations.

## Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* Addison-Wesley, 2011.

[2] Leffingwell, Dean. *Scaling Software Agility: Best Practices for Large Enterprises.* Addison-Wesley, 2007.

[3] Cohn, Mike. *Agile Estimating and Planning.* Robert C. Martin Series. Prentice Hall, 2005.