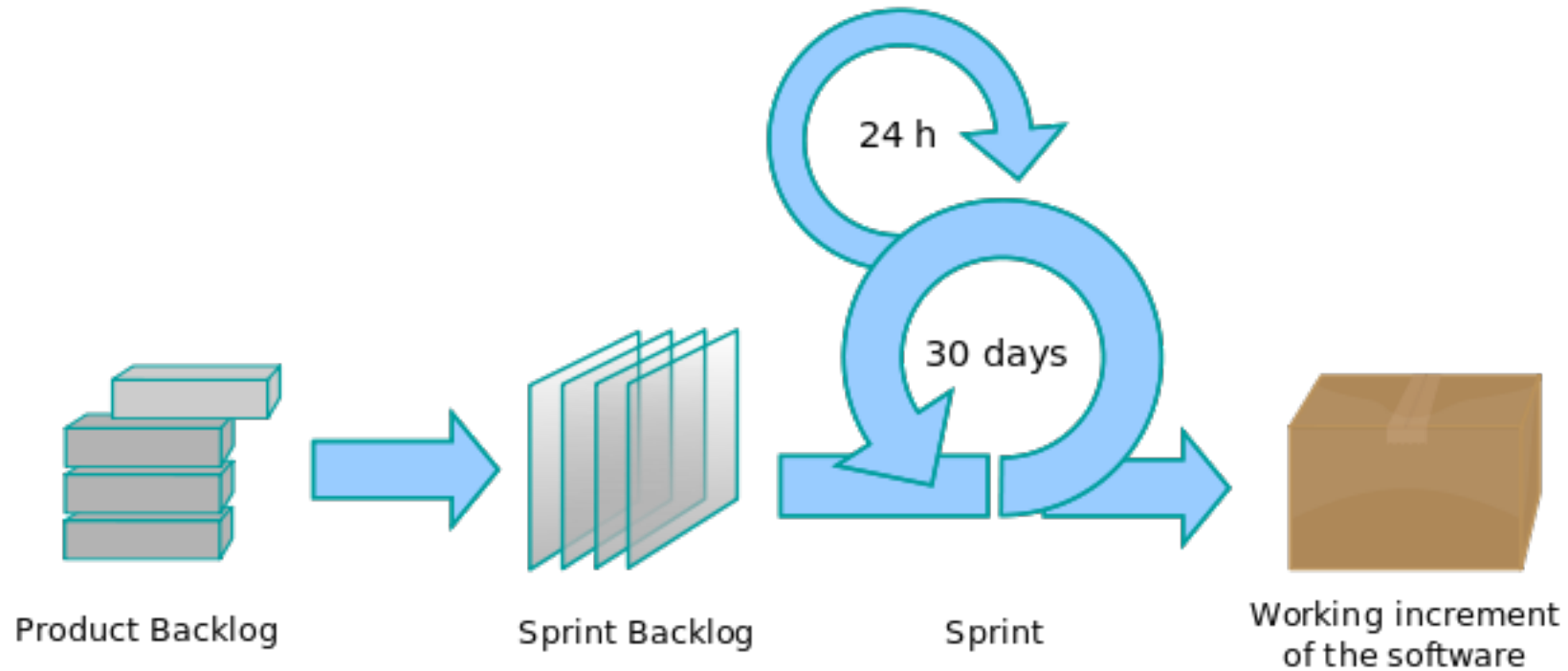


Sprint Planning



By Lakeworks - Own work, GFDL, <https://commons.wikimedia.org/w/index.php?curid=3526338>

SWEN-261

Introduction to Software Engineering

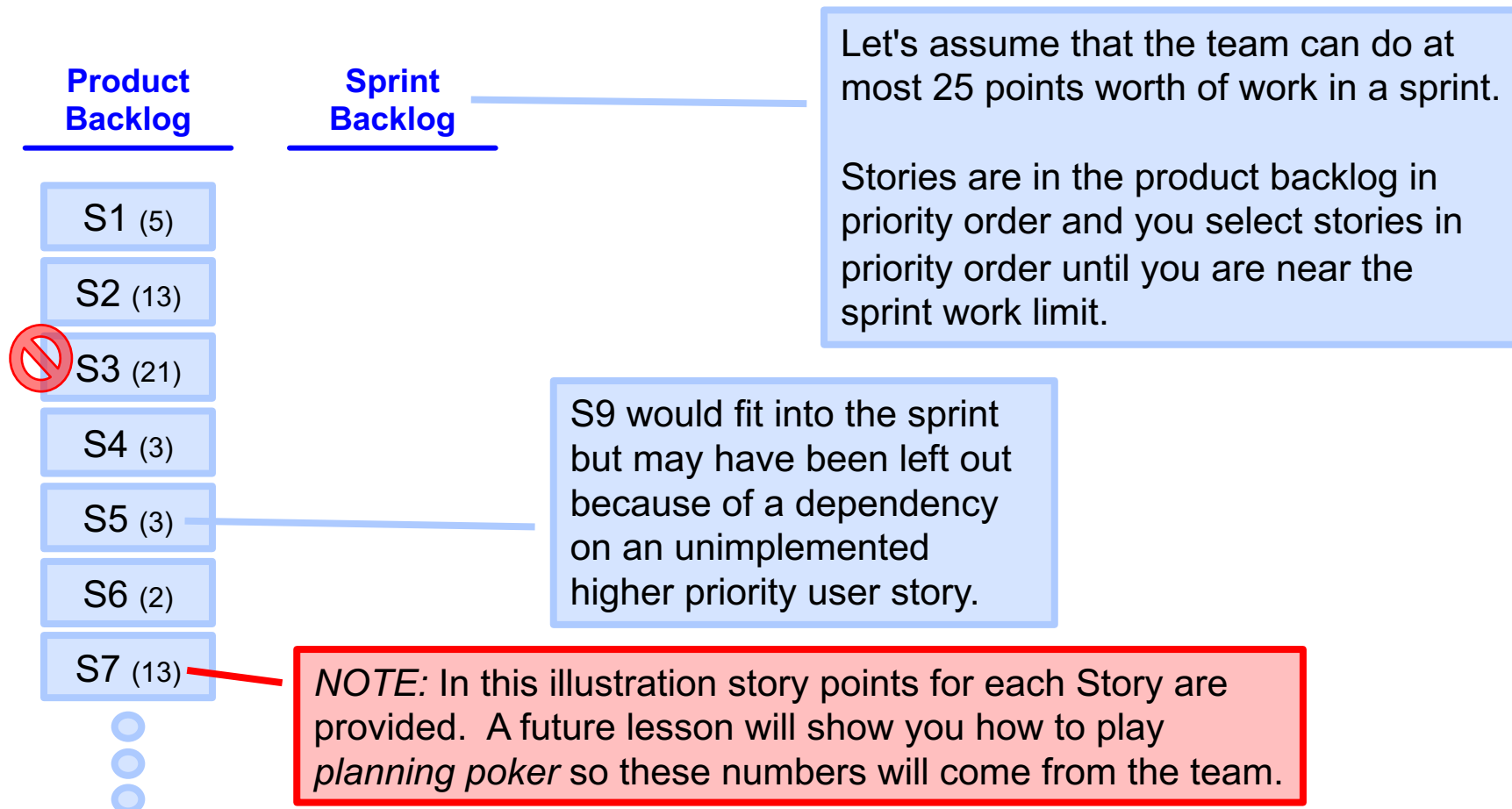
Department of Software Engineering
1 Rochester Institute of Technology

A sprint plan is a plan to build a small, working increment of the product.

- Each sprint is time-boxed; your sprints will be about three weeks long
- Each sprint has a backlog of work to be done during the sprint
 - ***This is the team's commitment to the Product Owner***
 - ***It is the sole focus on the team's effort during this sprint***
- The sprint planning meeting establishes the next sprint's backlog and launches the sprint
- Let's first consider how a sprint flows...

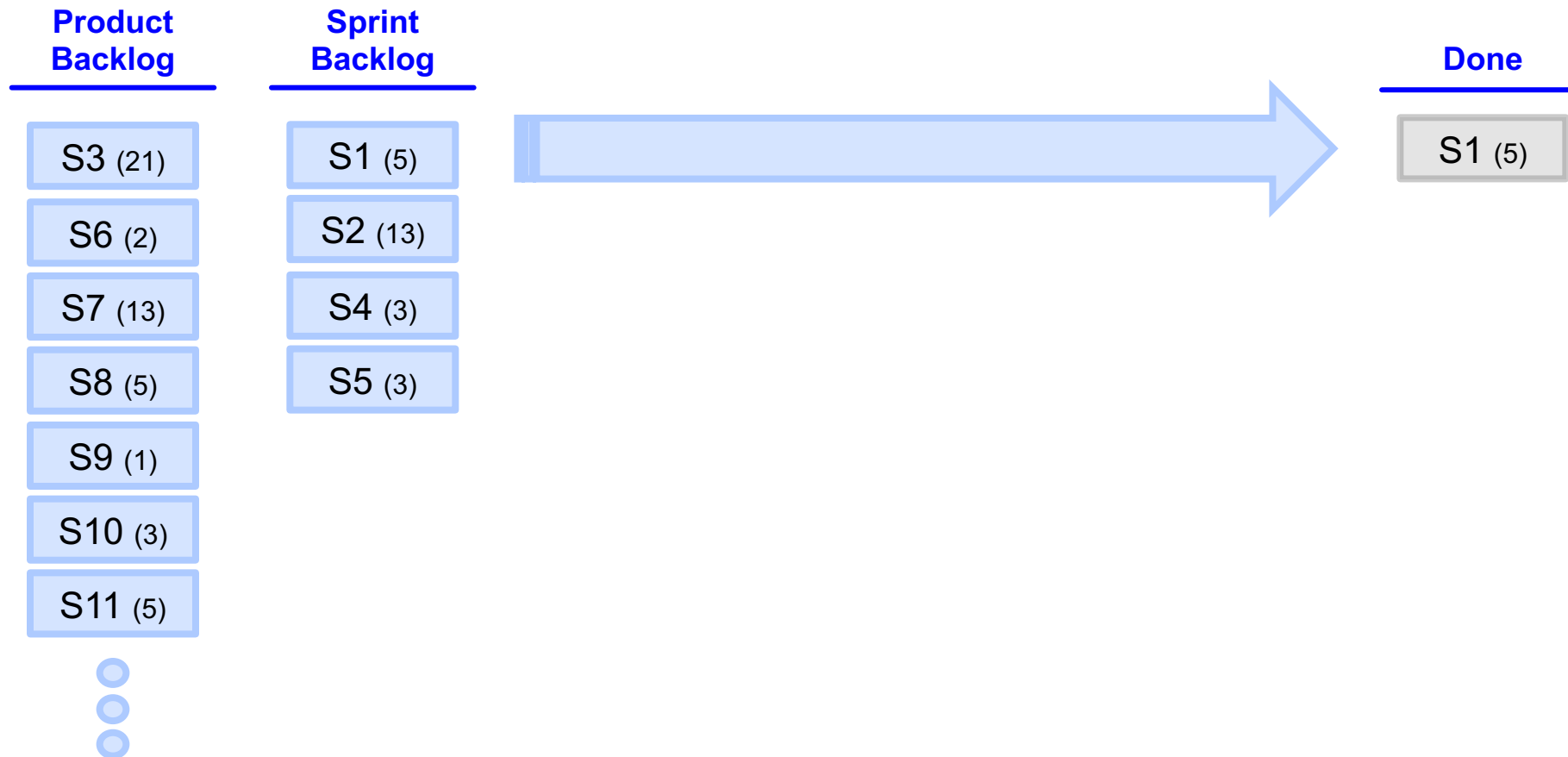
So we start with a backlog for the new sprint.

- During the Sprint Planning meeting, stories from the Product Backlog are put on the Sprint Backlog.



How does a story get from Backlog to Done?

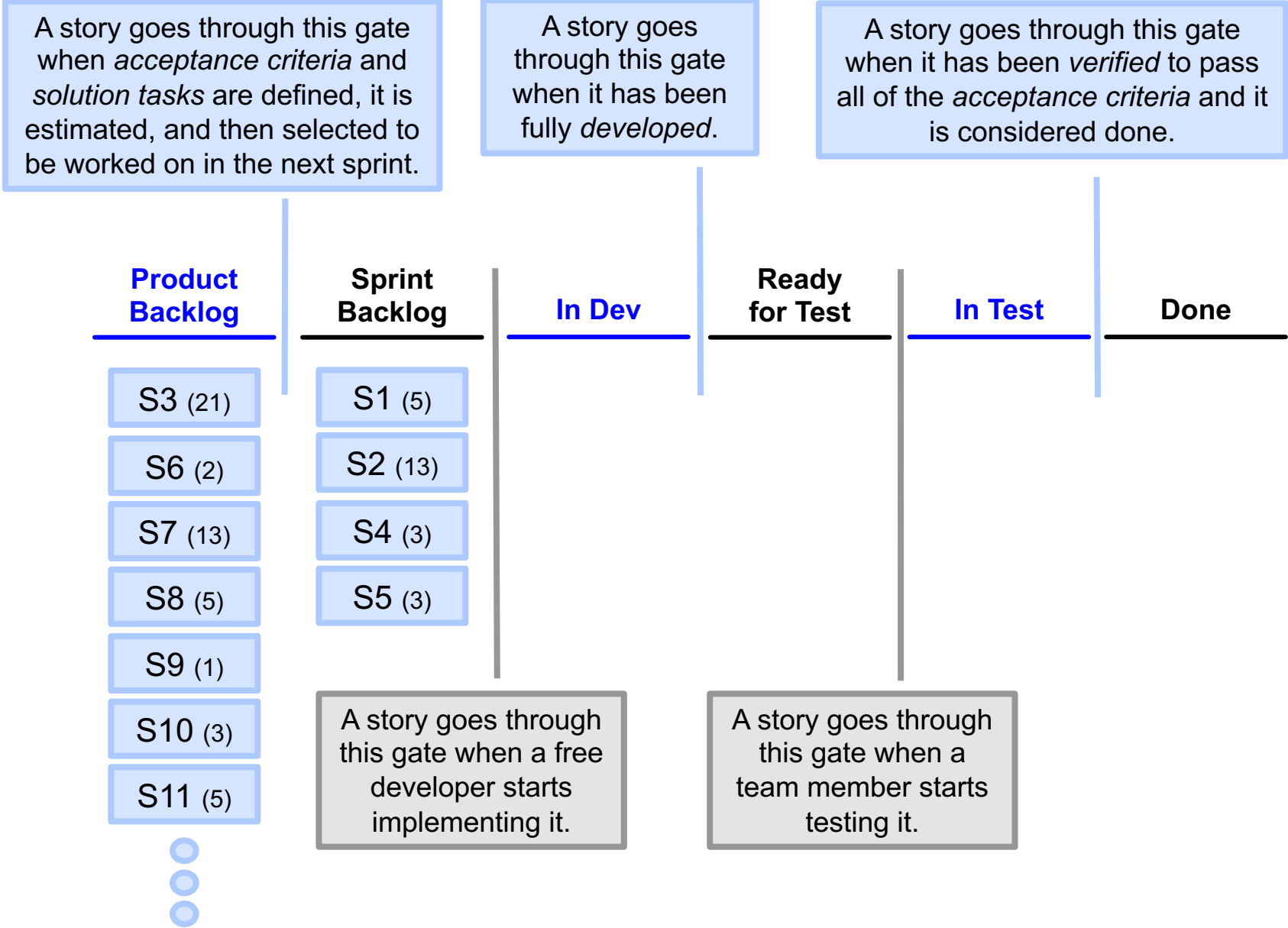
- When do you know that you're done with a story?



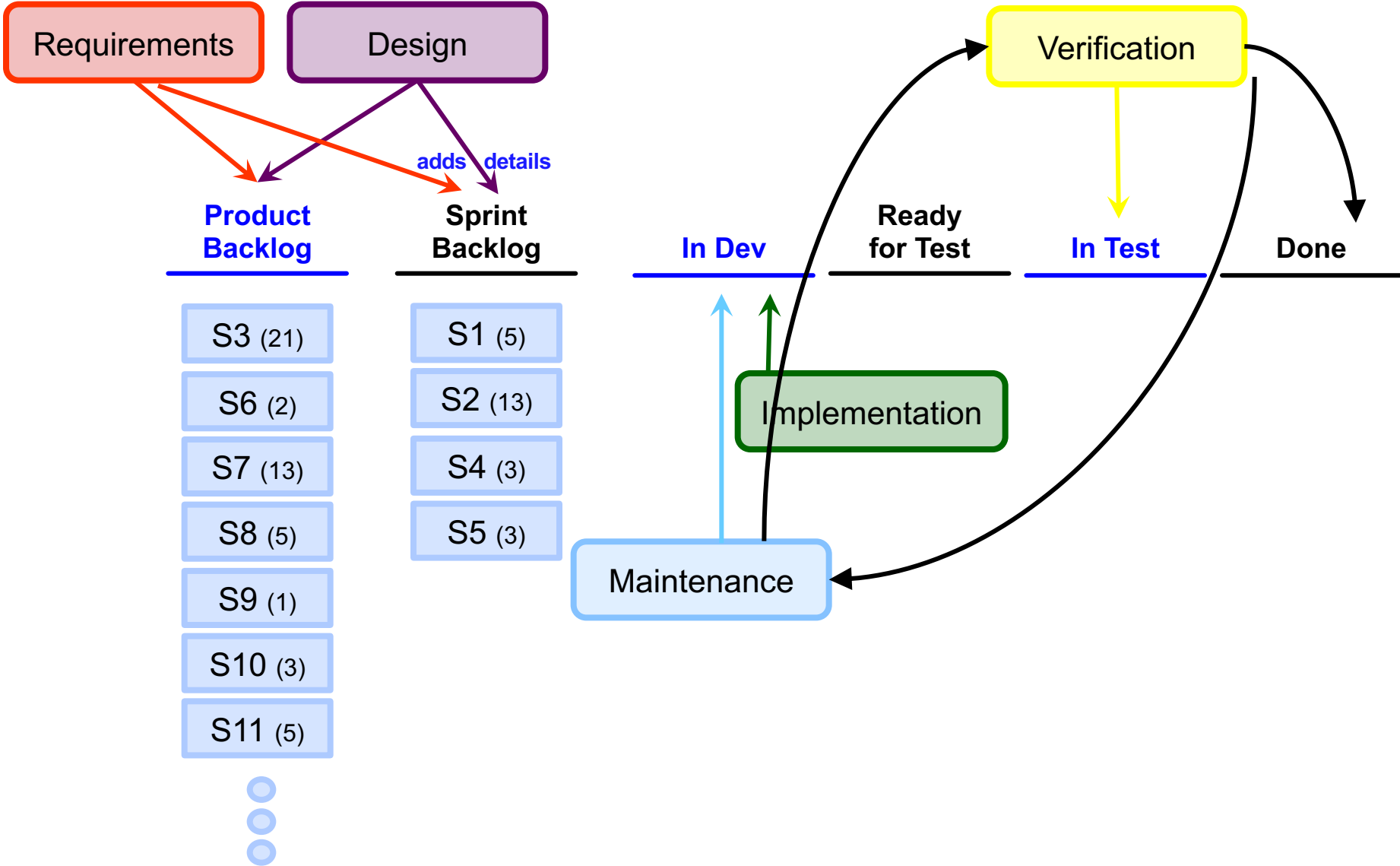
The team must establish a *Definition of Done*.

- Core activities:
 - *Acceptance criteria are defined*
 - *Solution is designed*
 - *Solution is developed; aka "code complete"*
 - *Feature has been tested (manually)*
- Other activities we'll add throughout the course:
 - *Feature branches (configuration management)*
 - *Unit testing*
 - *Code coverage analysis (goals)*
 - *Code reviews*
 - *Demo increment to stakeholders*

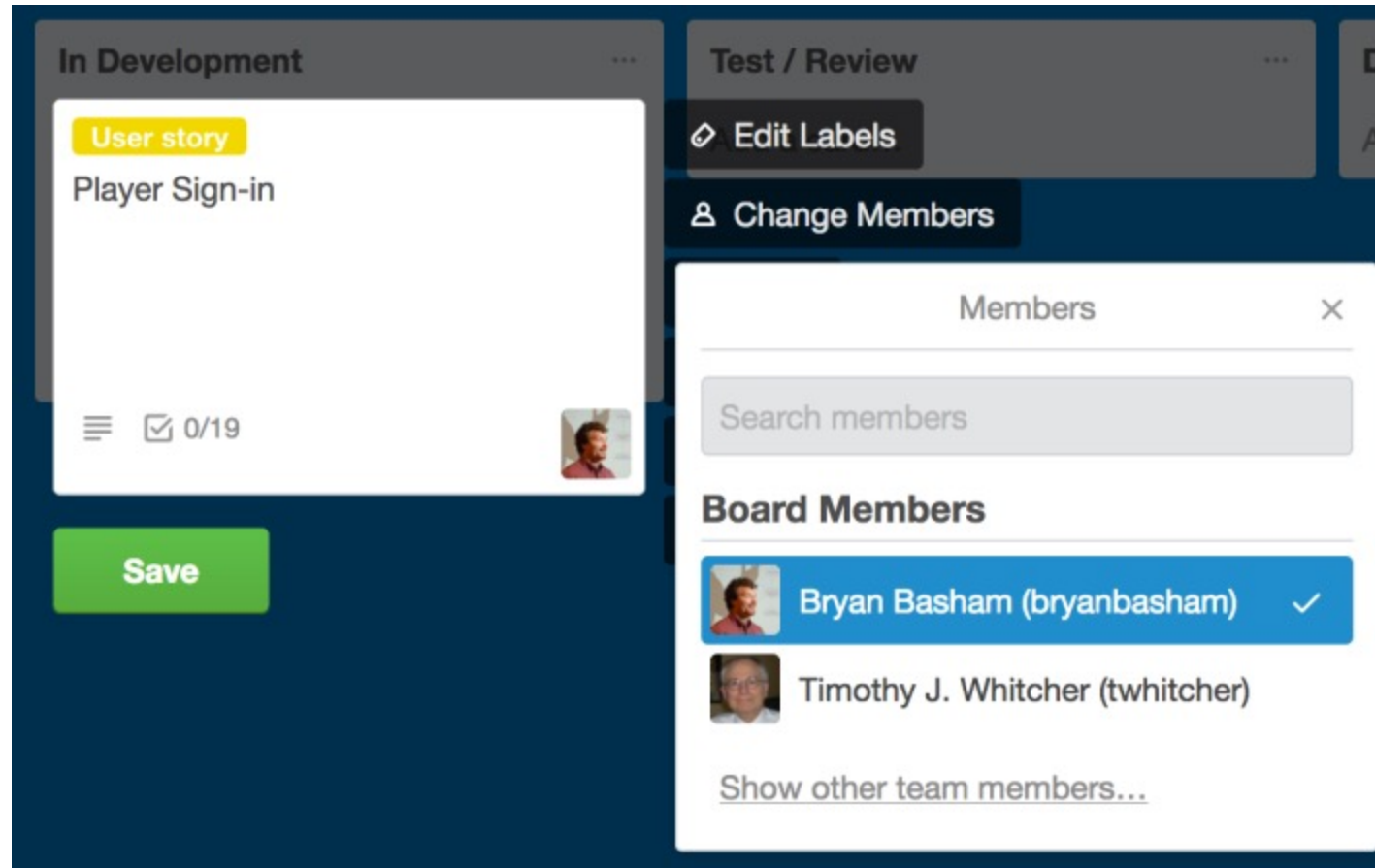
A story goes through five "gates" to get to *Done*.



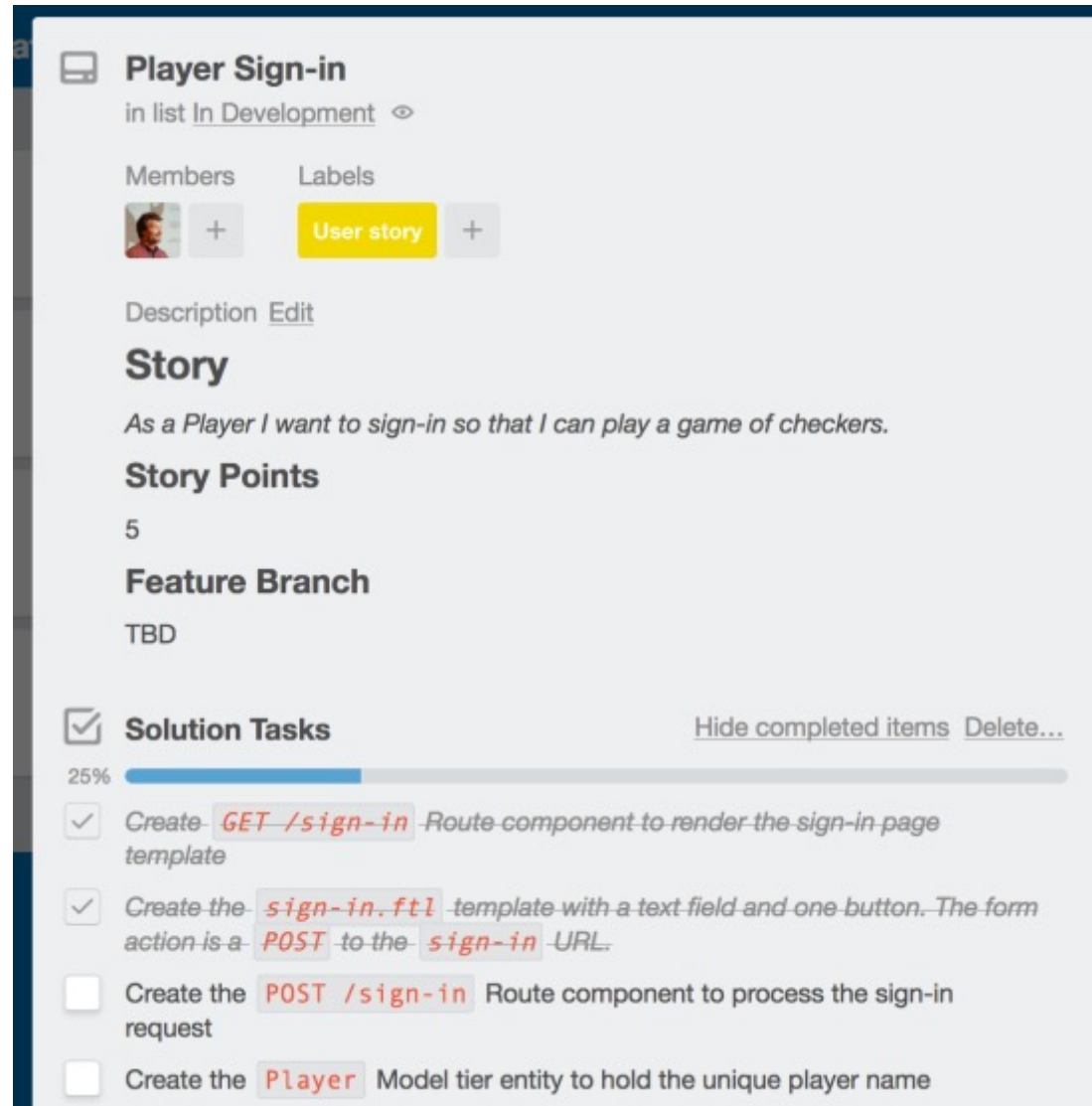
A sprint holds a mini-waterfall of activities



When members work on a Story they should be assigned to the card.



As solution tasks are completed, the person doing the work checks off the task as done.



The screenshot shows a Jira issue page for 'Player Sign-in'. The issue is in the 'In Development' list. It has one member and one label 'User story'. The description is 'As a Player I want to sign-in so that I can play a game of checkers.' The story points are 5, and the feature branch is TBD. Below the description is a 'Solution Tasks' section with a 25% progress bar. There are four tasks listed, with the first two checked off.

Player Sign-in
in list [In Development](#)

Members [+](#) Labels [User story](#) [+](#)

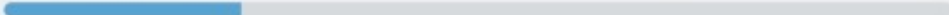
Description [Edit](#)

Story
As a Player I want to sign-in so that I can play a game of checkers.

Story Points
5

Feature Branch
TBD

Solution Tasks [Hide completed items](#) [Delete...](#)

25% 

- Create the `GET /sign-in` Route component to render the sign-in page template
- Create the `sign-in.ftl` template with a text field and one button. The form action is a `POST` to the `sign-in` URL.
- Create the `POST /sign-in` Route component to process the sign-in request
- Create the `Player` Model tier entity to hold the unique player name

Check-off *Definition of Done* items as the work gets completed.

Player Sign-in
in list [In Development](#)

Members + Labels **User story** +

Description [Edit](#)

Story
As a Player I want to sign-in so that I can play a game of checkers.

Story Points
5

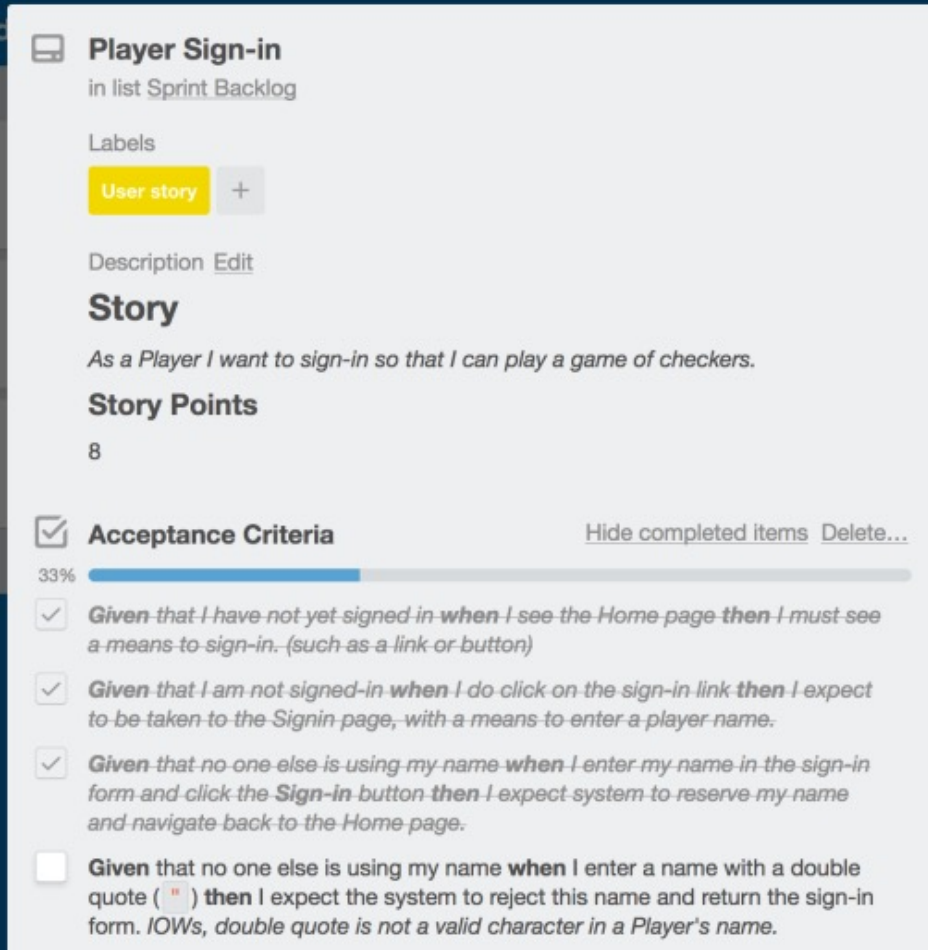
Feature Branch
TBD

Definition of Done Checklist [Hide completed items](#) [Delete...](#)

33%

- code-complete*
- manually test the story
- prepare for the feature demo

The tester checks off *Acceptance Criteria* as they validate the behavior of the system.



Player Sign-in
in list Sprint Backlog

Labels
User story +

Description [Edit](#)

Story

As a Player I want to sign-in so that I can play a game of checkers.

Story Points

8

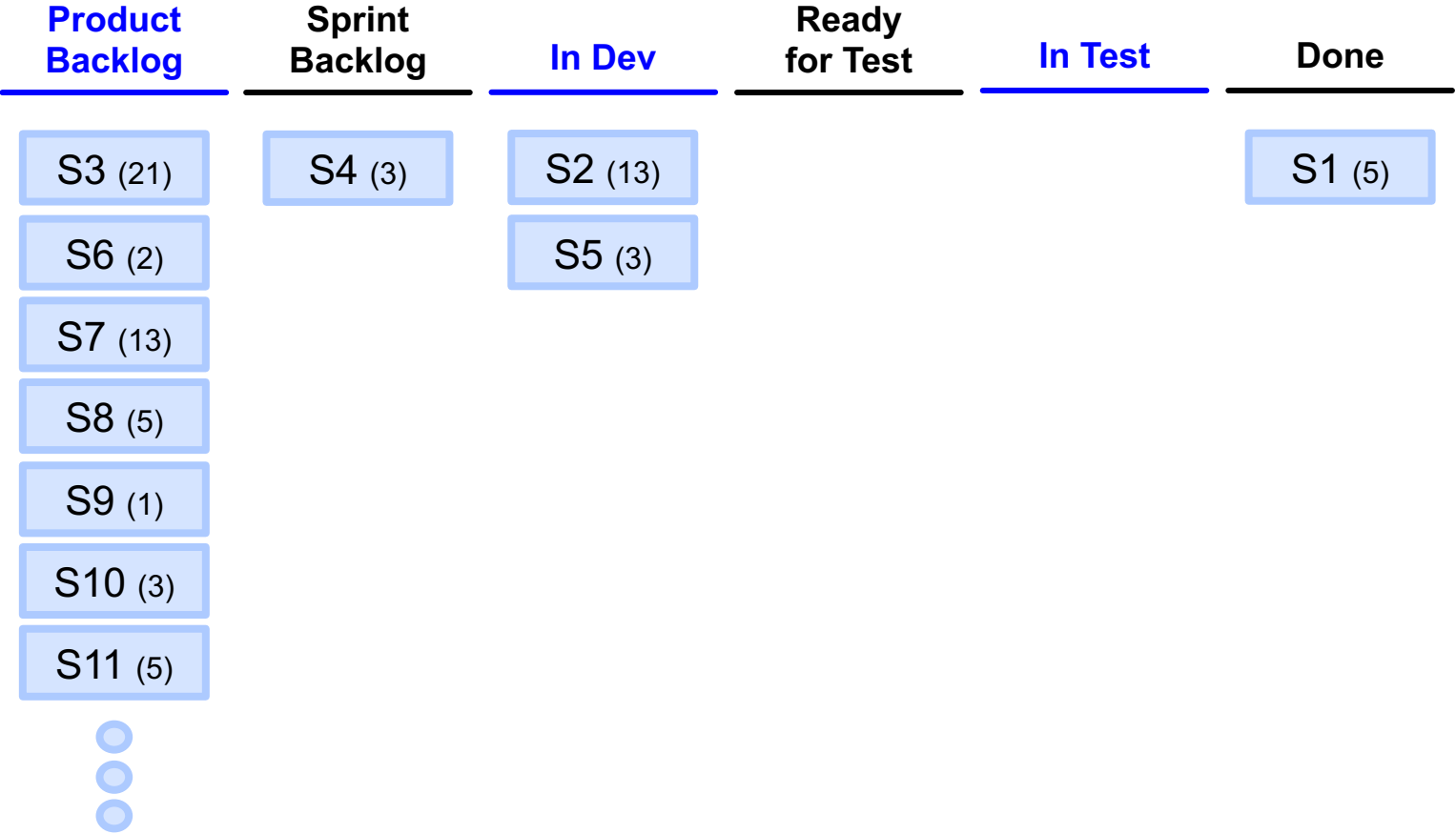
Acceptance Criteria [Hide completed items](#) [Delete...](#)

33%

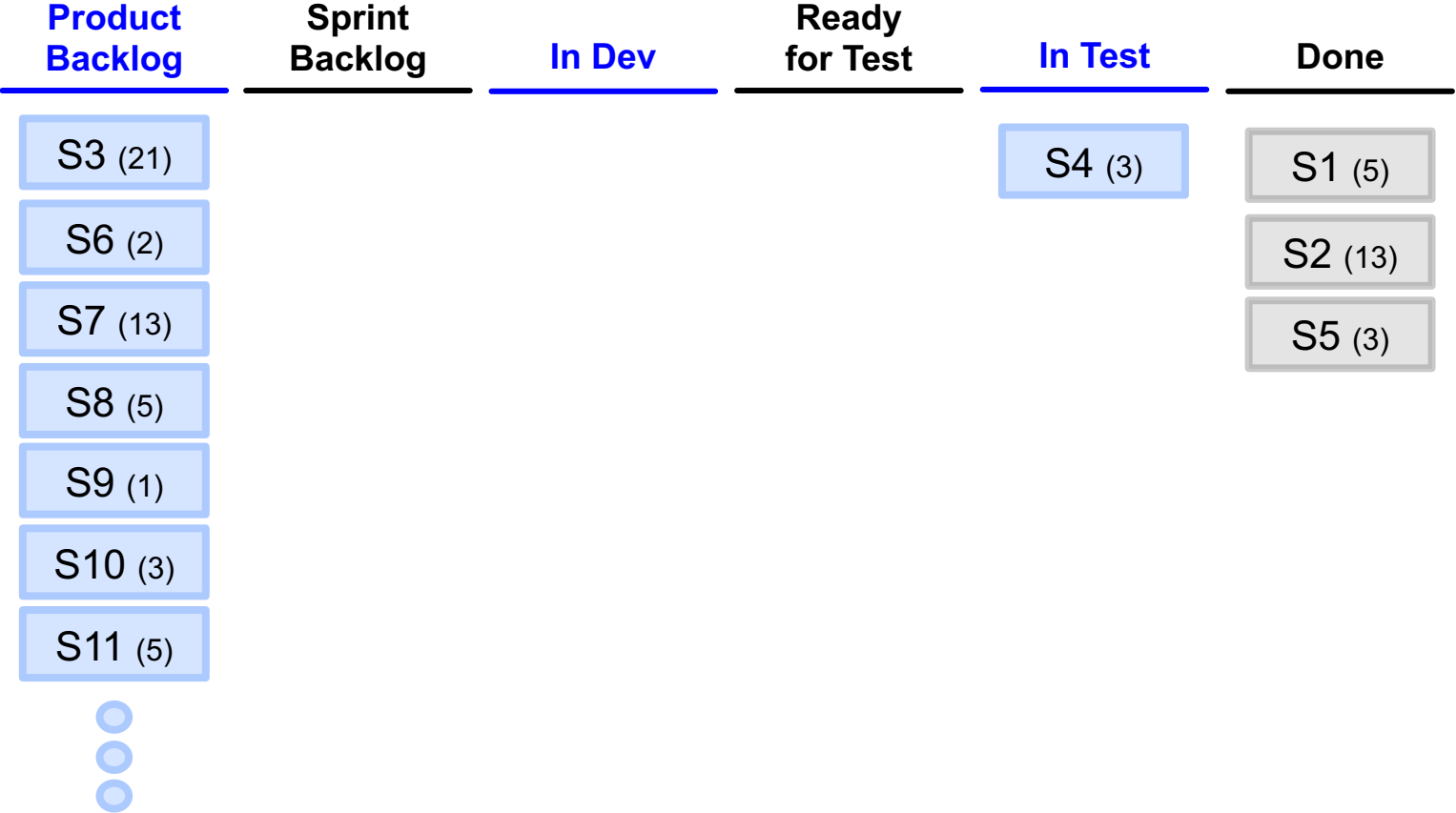
- Given** that I have not yet signed-in **when** I see the Home page **then** I must see a means to sign-in. (such as a link or button)
- Given** that I am not signed-in **when** I do click on the sign-in link **then** I expect to be taken to the Signin page, with a means to enter a player name.
- Given** that no one else is using my name **when** I enter my name in the sign-in form and click the **Sign-in** button **then** I expect system to reserve my name and navigate back to the Home page.
- Given** that no one else is using my name **when** I enter a name with a double quote (") **then** I expect the system to reject this name and return the sign-in form. IOWs, double quote is not a valid character in a Player's name.

A user story must pass its acceptance tests to be considered done.

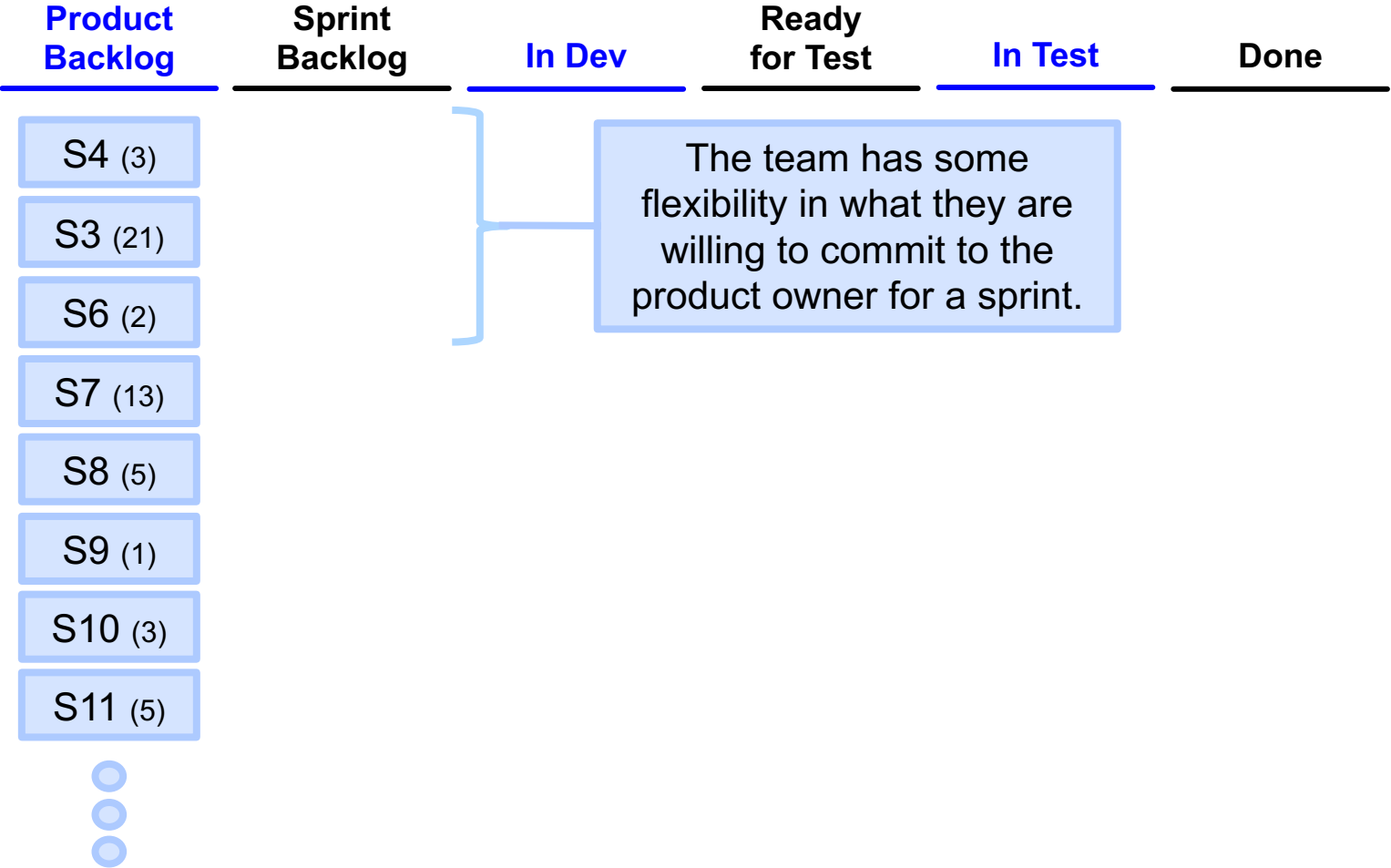
On any given day, the sprint board tells us the status of all sprint work.



At the end of a sprint, unfinished work gets put back onto the Product Backlog.



And the process starts all over again at the start of the next sprint.



But how do you really know how many story points the team can accomplish in a sprint?

- Velocity is measured from the average of the last three sprints.
 - *Only use the completed stories.*
 - *This rolling average will change over time.*
- The average velocity is then used to cap the number of story points for the next sprint.
- Example:
 - *Sprint 7: 45 story points committed; 42 completed.*
 - *Sprint 8: 40 committed; 50 completed*
 - *Sprint 9: 48 committed; 47 completed*
 - *Velocity = $(42 + 50 + 47) / 3 \approx 46.3333$*
 - *Sprint 10 will be capped at 46 story points.*

Velocity is specific to one team working on one project.

- This assumes that sprint length and team membership remain consistent.
 - *If either of these two change, then velocity measurement must start over with a new running average.*
- Velocity is only measured for a single project, single team.
 - *Story points are level of effort estimates*
 - *Story points are determined by the team, for the team*
 - *Thus you cannot compare velocity's between teams*
- Management cannot **set** a team's velocity.

Velocity is not the same as the team's overall capacity.

- Velocity is only a measure of effort for working on user stories.
 - *It does not include company meetings, email communication, small "outside" tasks*
 - *For class, it does not include most before- and after-class activities*
- What about these issues?
 - *Holidays or vacations*
 - *Members given large, outside tasks*
- Normally these issues can be ignored being smoothed out by the averaging process.
- In extreme cases, the team can make adjustments to the calculated velocity usually by lowering it.