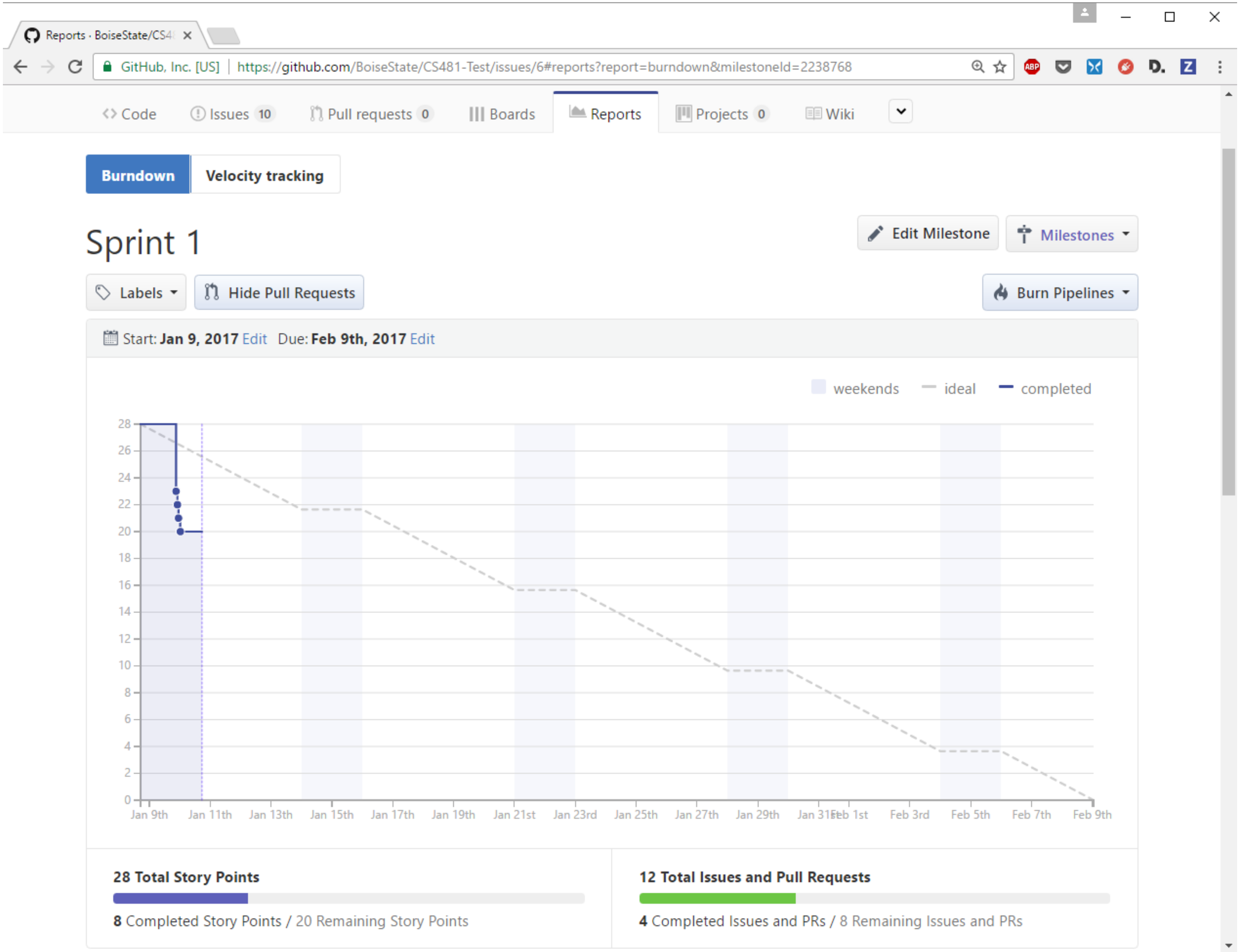
Example Sprint Burndown Chart (GitHub + ZenHub)



Sprint 1

Start: **Jan 9, 2017** Edit Due: **Feb 9th, 2017** Edit

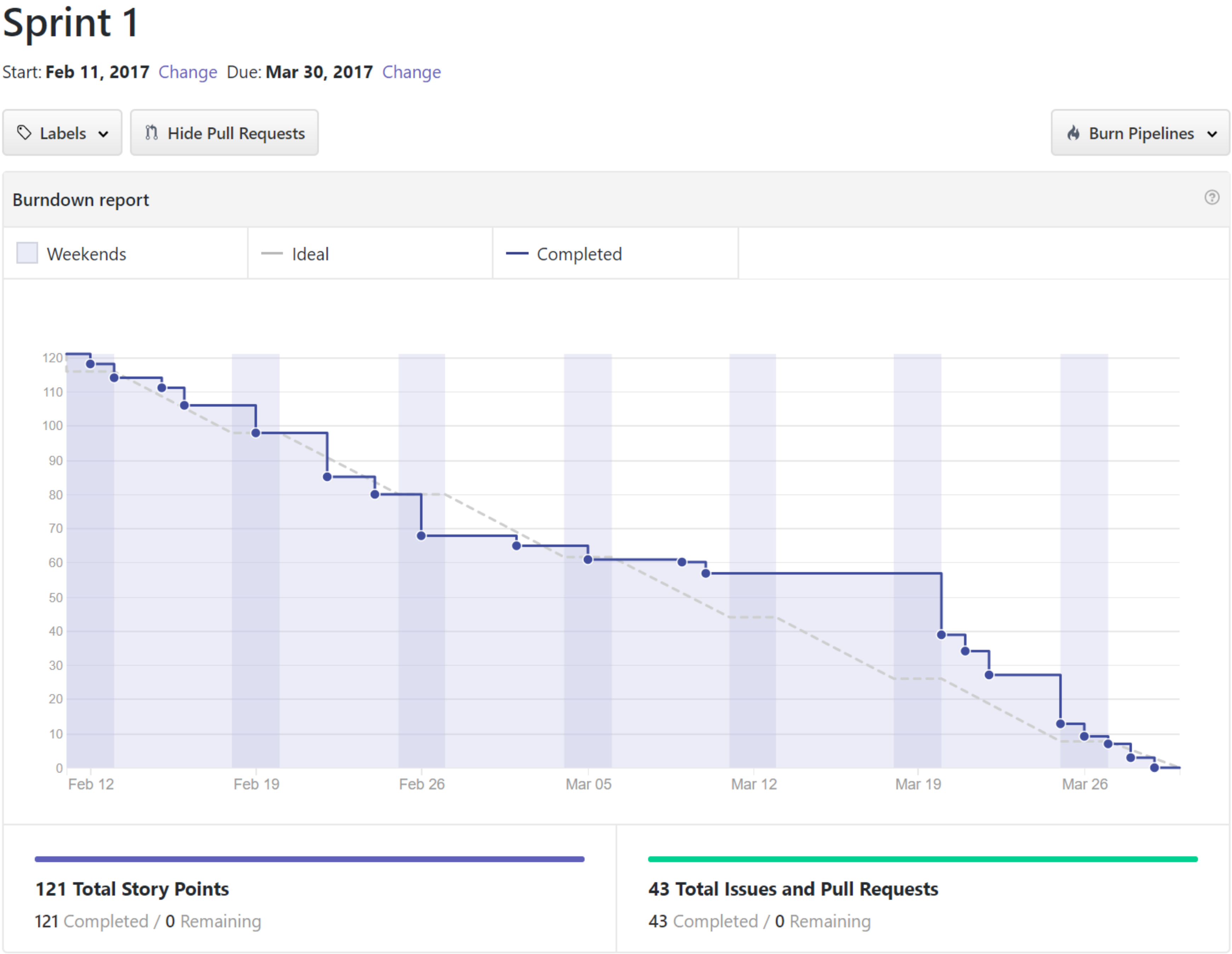
28 Total Story Points

8 Completed Story Points / 20 Remaining Story Points

12 Total Issues and Pull Requests

4 Completed Issues and PRs / 8 Remaining Issues and PRs

Example Sprint Burndown Chart (GitHub + ZenHub)



Sprint 1

Start: **Feb 11, 2017** Change Due: **Mar 30, 2017** Change

Burndown report

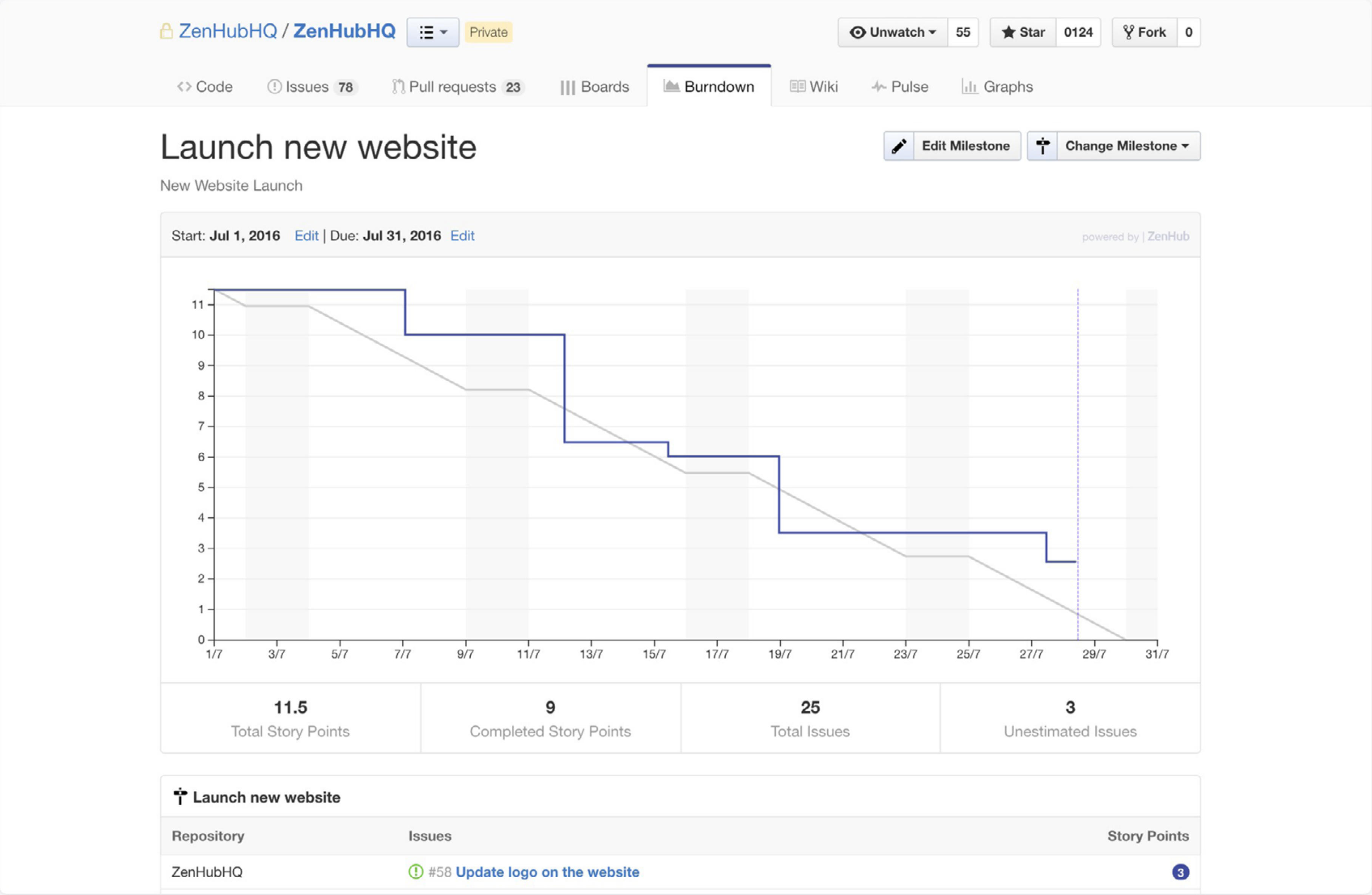
121 Total Story Points

121 Completed / 0 Remaining

43 Total Issues and Pull Requests

43 Completed / 0 Remaining

Example Sprint Burndown Chart (GitHub + ZenHub)



Launch new website

New Website Launch

Start: **Jul 1, 2016** Edit | Due: **Jul 31, 2016** Edit

11.5	9	25	3
Total Story Points	Completed Story Points	Total Issues	Unestimated Issues

Launch new website

Repository	Issues	Story Points
ZenHubHQ	#58 **Update logo on the website**	3

Scrum Artifacts: Definition of Done

Scrum Artifacts: Definition of Done

- What is it?

- Who defines it?

Scrum Artifacts: Definition of Done

- Scrum teams define the conditions necessary for a User Story to be considered “done”

- The definition of “done” may vary from one project to the next. Why?

Scrum Artifacts: Definition of Done

- Scrum teams define the conditions necessary for a User Story to be considered “done”

- The definition of “done” may vary from one project to the next. Why?
  - Different types of projects (e.g., backend mostly vs. web/mobile development)
  - Different quality standards:
    - e.g., server needs to be more robust than front-end

Scrum Artifacts: Definition of Done

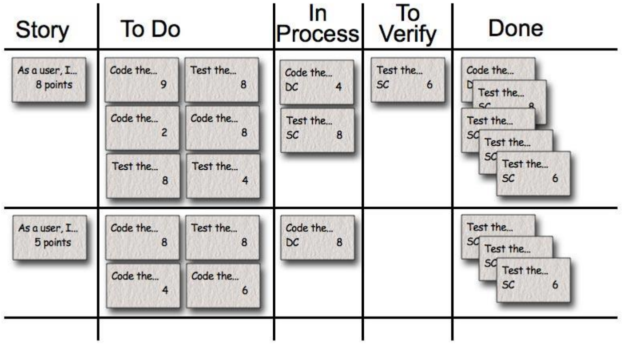
- Typical conditions for “done” in a software project include:
  - All Acceptance Criteria passing
  - All unit-level tests passing
  - Completion of all engineering practices (static analysis, peer reviews, test coverage goals,…)

Scrum Artifacts: Definition of Done

- Our CS471 projects will have a more complete definition of “done”
  - refactoring
  - performance testing
  - documentation updated
  - does the customer needs to sign off on a feature/story?
  - does the feature/story have to be beta-tested?
  - etc.

Scrum Artifacts Review

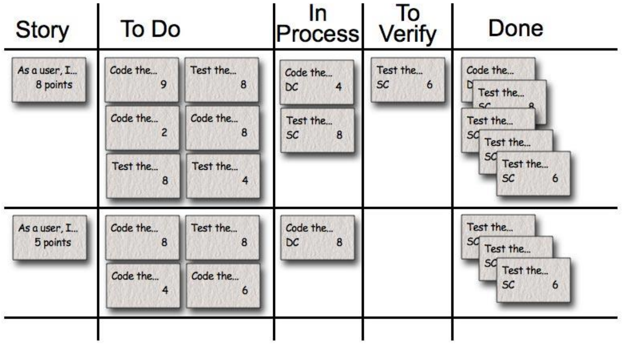
- What features are we working on now and how do we coordinate our work?
  - How will we implement those features?
  - Who is or will do the work?



Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6

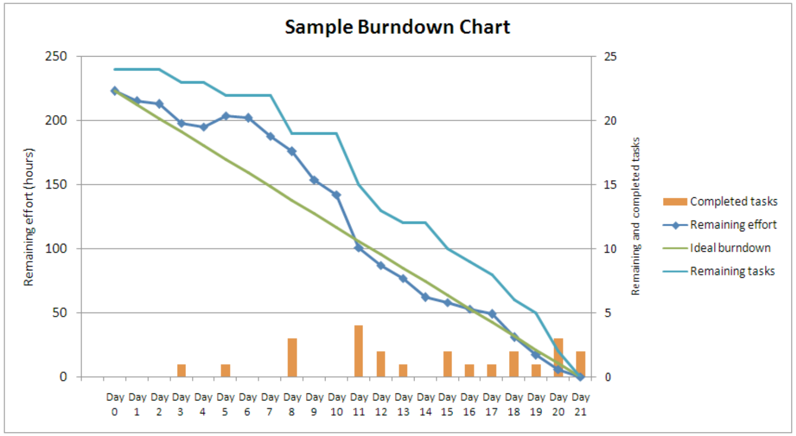
Scrum Artifacts Review

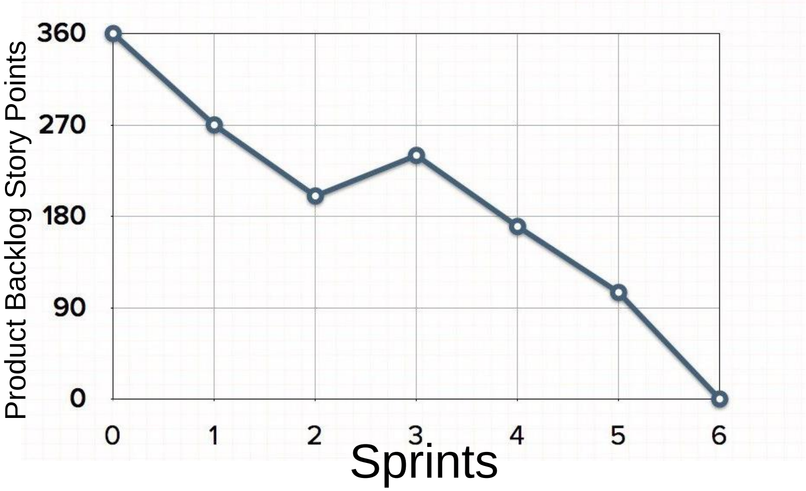
- What features are we working on now and how do we coordinate our work?
  - How will we implement those features?
  - Who is or will do the work?



Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8 Code the... 2 Code the... 8 Test the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8 Test the... 8 Code the... 4 Code the... 6	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6

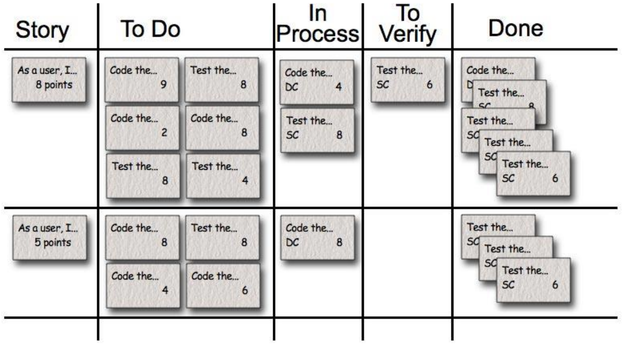
- When will we finish this sprint? ...everything?





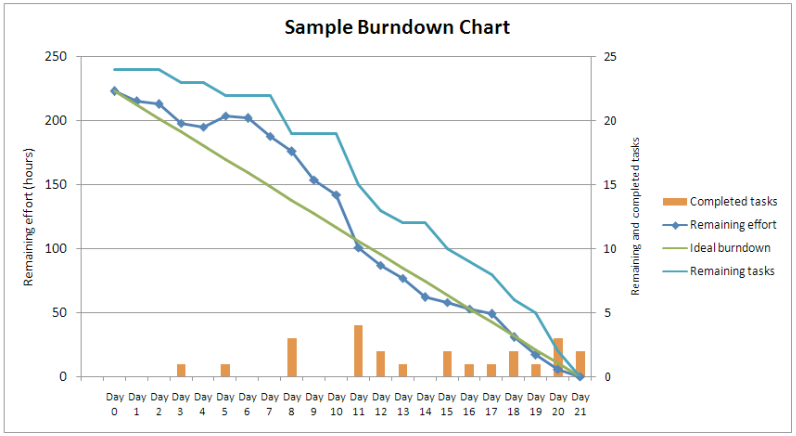
Scrum Artifacts Review

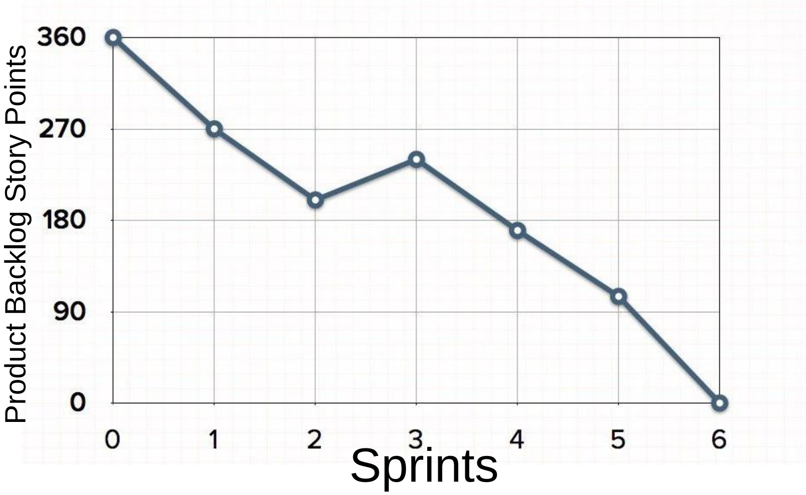
- What features are we working on now and how do we coordinate our work?
  - How will we implement those features?
  - Who is or will do the work?



Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8 Code the... 2 Code the... 8 Test the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8 Test the... 8 Code the... 4 Code the... 6	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6

- When will we finish this sprint? ...everything?



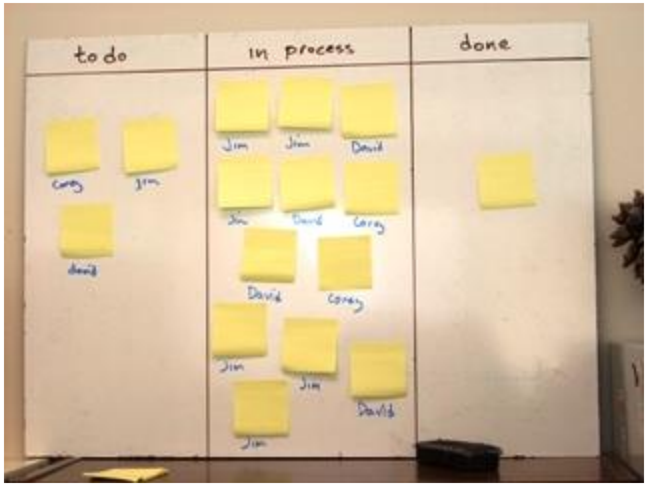


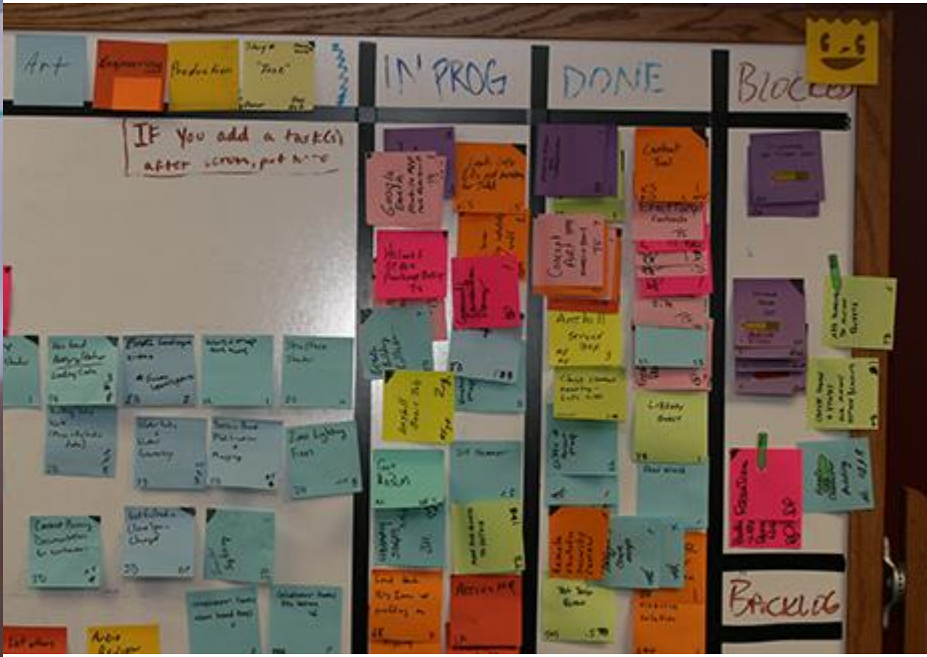
- What is meant by done or completed?

Scrum Tools

Scrum Tools

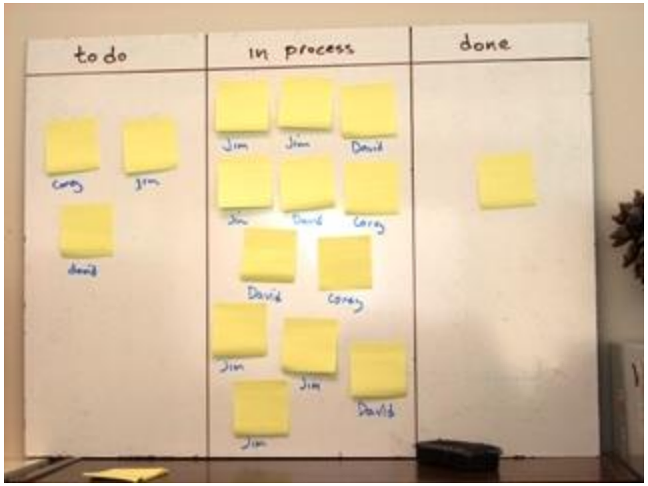
- Which teams can use this?

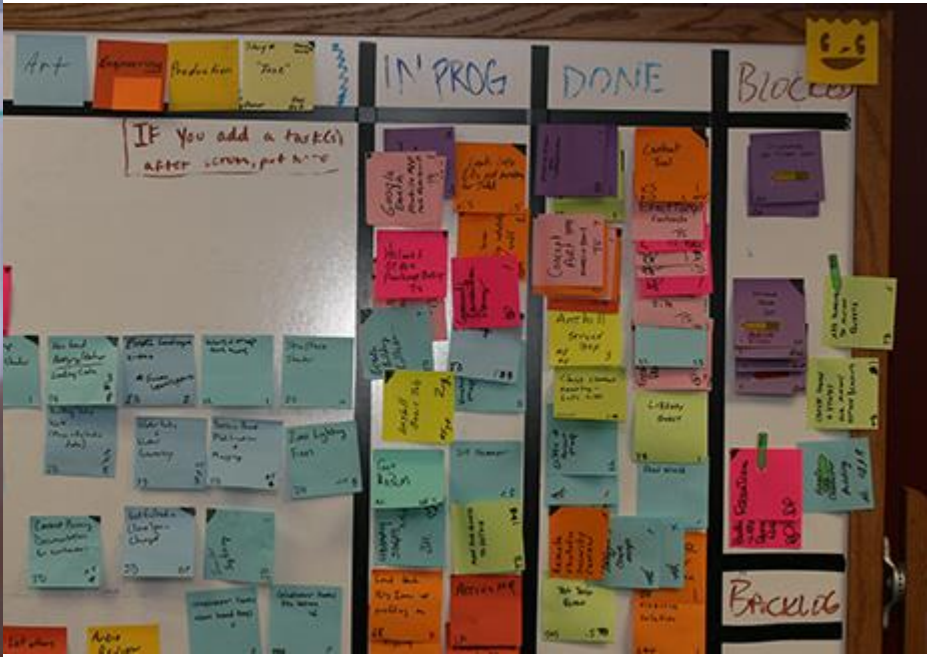




Scrum Tools

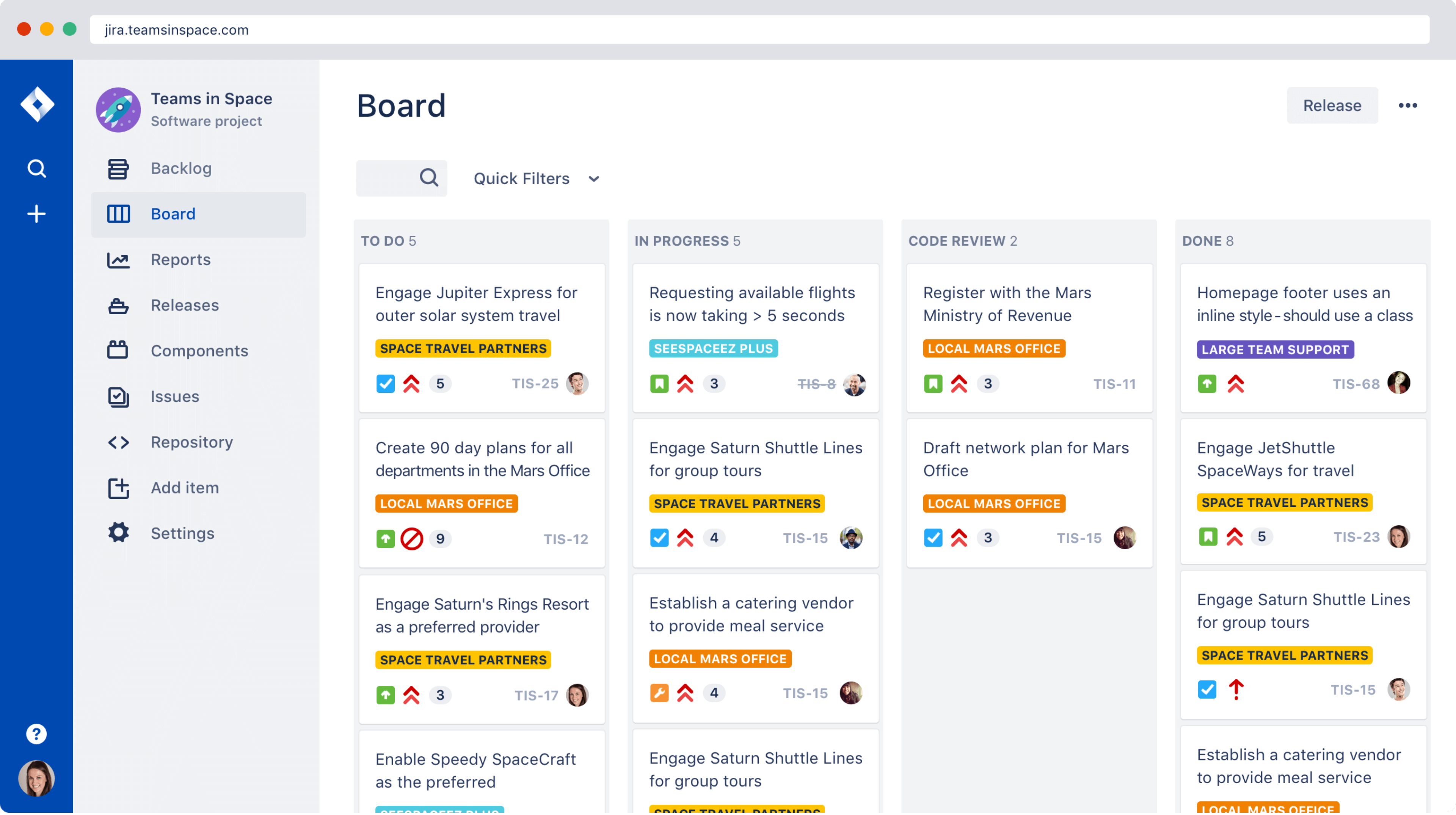
- Artifacts may simply reside as Post-It Notes on Whiteboards for co-located teams





Scrum Tools (Jira)

- On-line for non-co-located teams



Board

TO DO 5

Engage Jupiter Express for outer solar system travel
SPACE TRAVEL PARTNERS
5 TIS-25

Create 90 day plans for all departments in the Mars Office
LOCAL MARS OFFICE
9 TIS-12

Engage Saturn's Rings Resort as a preferred provider
SPACE TRAVEL PARTNERS
3 TIS-17

Enable Speedy SpaceCraft as the preferred
SEESPACEEZ PLUS

IN PROGRESS 5

Requesting available flights is now taking > 5 seconds
SEESPACEEZ PLUS
3 ~~TIS-8~~

Engage Saturn Shuttle Lines for group tours
SPACE TRAVEL PARTNERS
4 TIS-15

Establish a catering vendor to provide meal service
LOCAL MARS OFFICE
4 TIS-15

Engage Saturn Shuttle Lines for group tours
SPACE TRAVEL PARTNERS

CODE REVIEW 2

Register with the Mars Ministry of Revenue
LOCAL MARS OFFICE
3 TIS-11

Draft network plan for Mars Office
LOCAL MARS OFFICE
3 TIS-15

DONE 8

Homepage footer uses an inline style - should use a class
LARGE TEAM SUPPORT
TIS-68

Engage JetShuttle SpaceWays for travel
SPACE TRAVEL PARTNERS
5 TIS-23

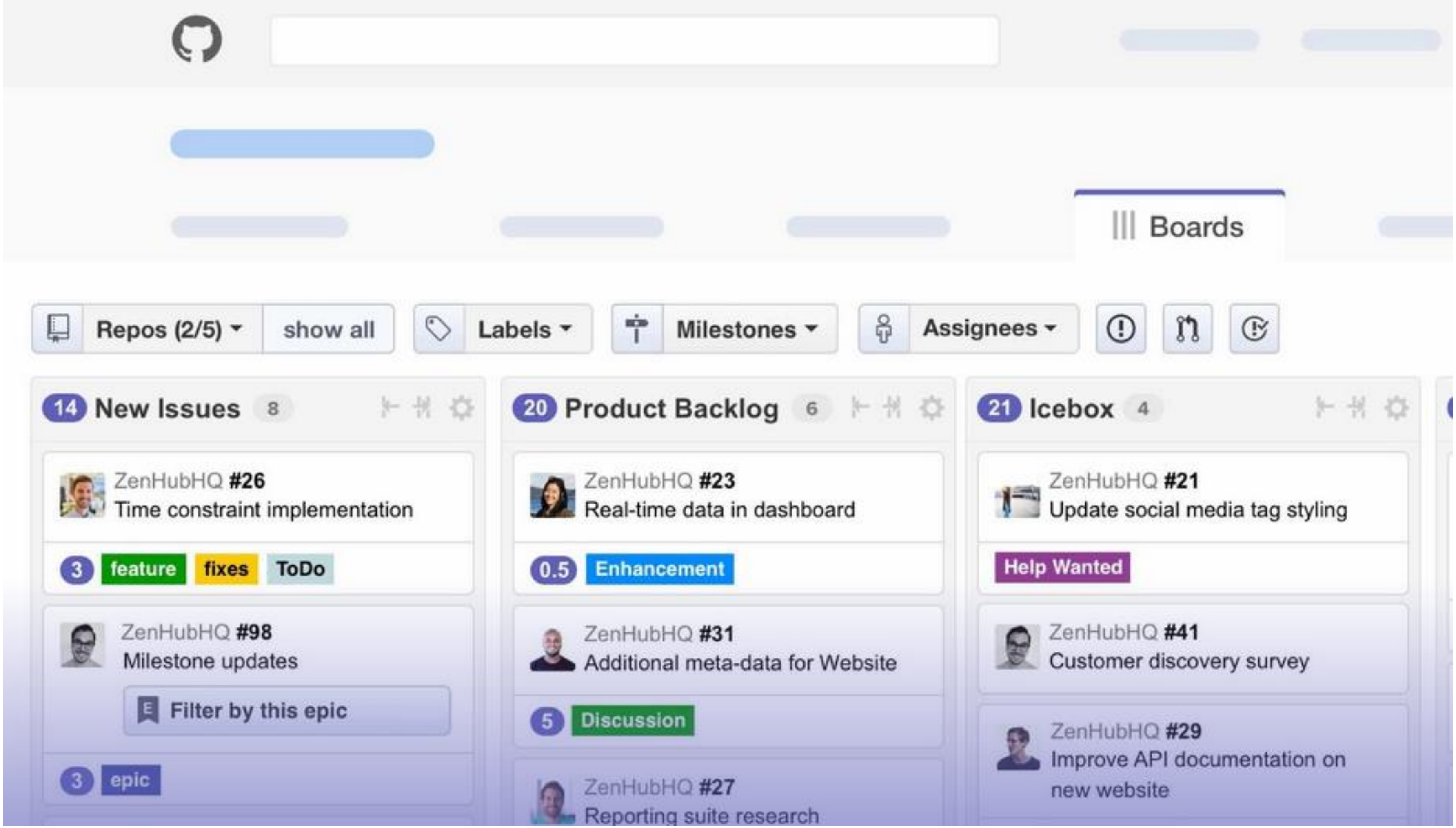
Engage Saturn Shuttle Lines for group tours
SPACE TRAVEL PARTNERS
TIS-15

Establish a catering vendor to provide meal service
LOCAL MARS OFFICE

https://www.atlassian.com/software/jira

Scrum Tools CS471/CS481: ZenHub (integrated with GitHub)

- Switched from using Google Spreadsheets in Fall'16



https://www.zenhub.com/

Additional Cloud-Based Agile Project Management Tools

- Team Foundation Server/Service (TFS) https://www.visualstudio.com/team-services/
- https://trello.com/
- https://www.versionone.com/
- https://www.agilefant.com/
- https://www.yodiz.com/
- https://www.ca.com/
- etc., google “cloud agile project management”

Scrum: Sprint

Scrum: Sprint

- A *sprint* defines the team’s activities in each iteration
- A sprint is *timeboxed* (typical duration: 1..4 weeks)
- Each sprint releases a potentially shippable (fully tested) increment of functionality

Sprint Property: Timeboxed

- The duration of each sprint is established before it begins
- At the end of the timebox, the sprint releases the completed user stories
- Incomplete user stories …

Sprint Property: Timeboxed

- The duration of each sprint is established before it begins
- At the end of the timebox, the sprint releases the completed user stories
- Incomplete user stories are *snow plowed* into a future sprint
- *Availability* becomes a requirement competing with other requirements (features, quality, performance…)

Snow Plowing

- Incomplete work pushed into a growing pile to do “someday”
- Reasons for snow plowing:

Snow Plowing

- Incomplete work pushed into a growing pile to do “someday”
- Reasons for snow plowing:

Snow Plowing

- Incomplete work pushed into a growing pile to do "someday"

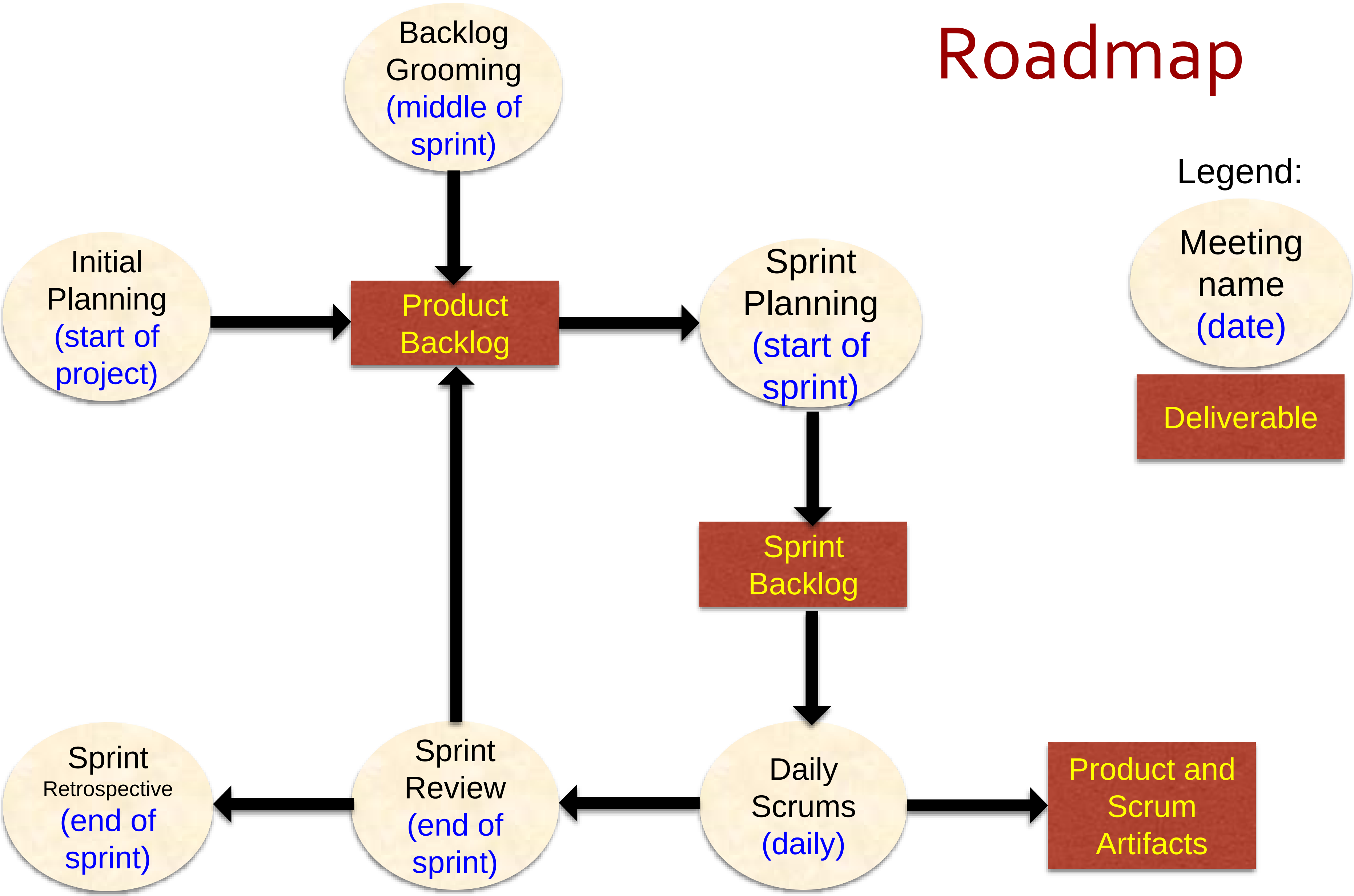
- Reasons for snow plowing:
  - Failed to divide a complex story into smaller stories
  - More difficult than expected due to unanticipated technical difficulty
  - More difficult than expected due to broken estimation process
  - Disaster strikes (hard disk failure, illness, flood, fire, cyber attack…)
  - Scrum Master fails to enforce Scrum rules

Snow Plowing

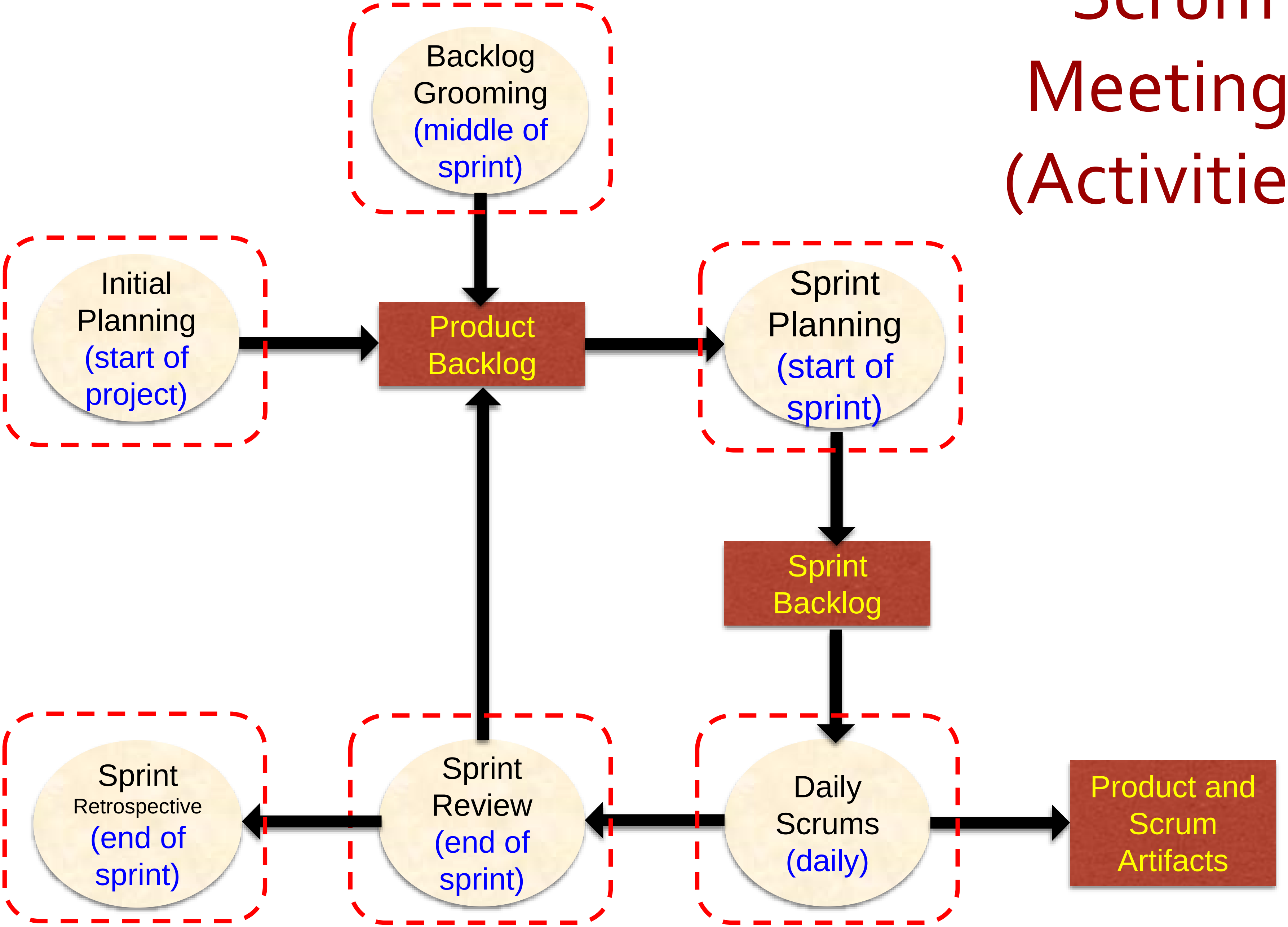
- Make snow plowed work visible — Never hide snow plowed work!

- Agile artifacts document “The current version of The Truth” :)

Scrum Process Roadmap


Backlog Grooming (middle of sprint)
Initial Planning (start of project)
Product Backlog
Sprint Planning (start of sprint)
Legend:
Meeting name (date)
Deliverable
Sprint Backlog
Sprint Retrospective (end of sprint)
Sprint Review (end of sprint)
Daily Scrums (daily)
Product and Scrum Artifacts

Scrum Meetings (Activities)



Backlog Grooming (middle of sprint)

Initial Planning (start of project)

Product Backlog

Sprint Planning (start of sprint)

Sprint Backlog

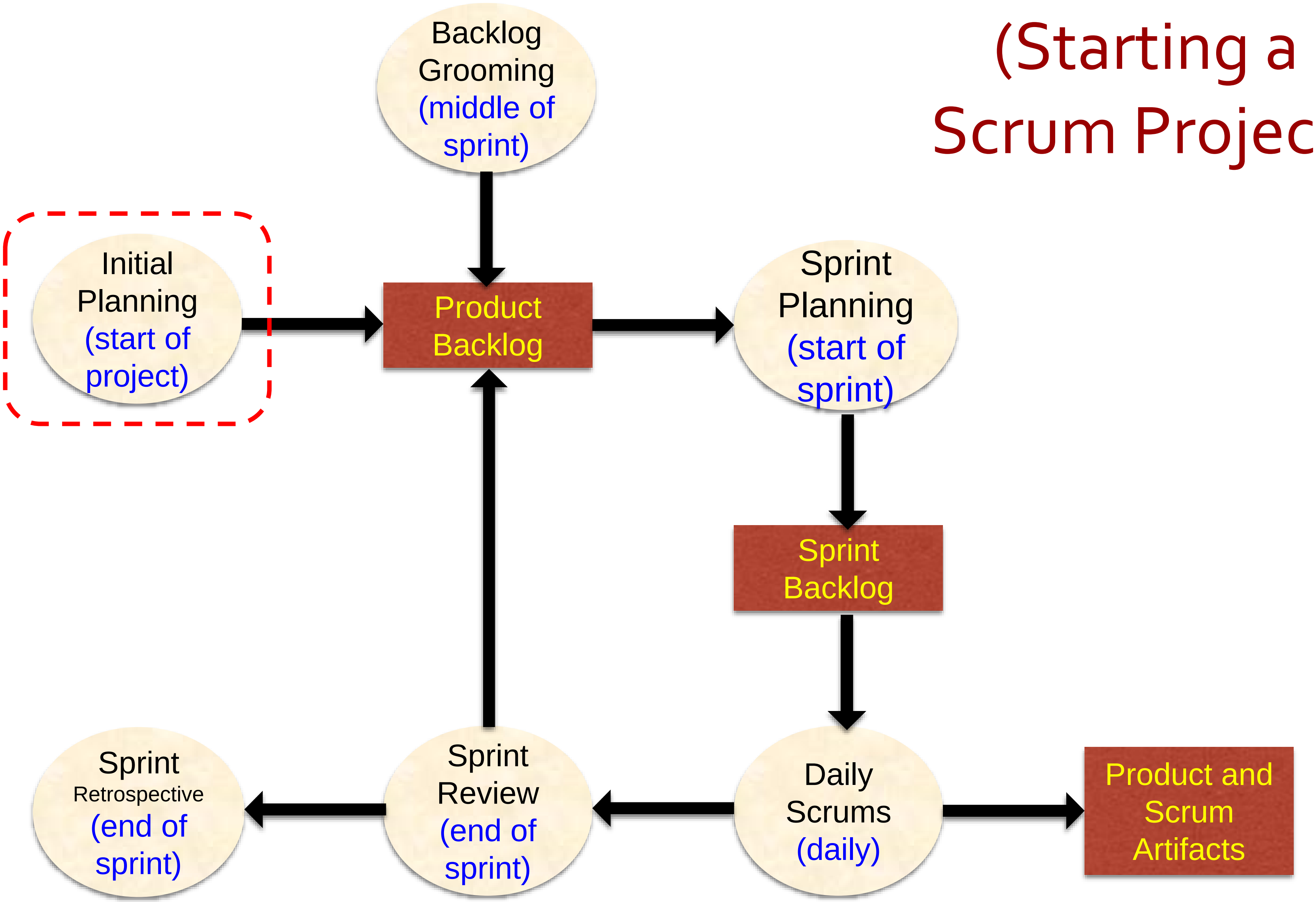
Daily Scrums (daily)

Product and Scrum Artifacts

Sprint Review (end of sprint)

Sprint Retrospective (end of sprint)

Initial Planning (Starting a Scrum Project)



Backlog Grooming (middle of sprint)
Initial Planning (start of project)
Product Backlog
Sprint Planning (start of sprint)
Sprint Backlog
Sprint Retrospective (end of sprint)
Sprint Review (end of sprint)
Daily Scrums (daily)
Product and Scrum Artifacts

Sprint Activities: Initial Planning Meeting

- An Initial Planning meeting creates the initial Product Backlog

- Note: Joining an existing team working on an ongoing project you may not participate in the Initial Planning activity

Sprint Activities: Initial Planning Meeting

- Who is responsible for the first draft of the Product Backlog?

Sprint Activities: Initial Planning Meeting

- Prior to the Initial Planning meeting, the Product Owner works with the customer/user to create the “first draft” of the Product Backlog, by:
  - Writing the initial User Stories
  - Writing basic Acceptance Criteria for each user story
  - Prioritizing the initial User Stories

Sprint Activities: Initial Planning Meeting

- The whole team meets during the Initial Planning meeting to:

Sprint Activities: Initial Planning Meeting

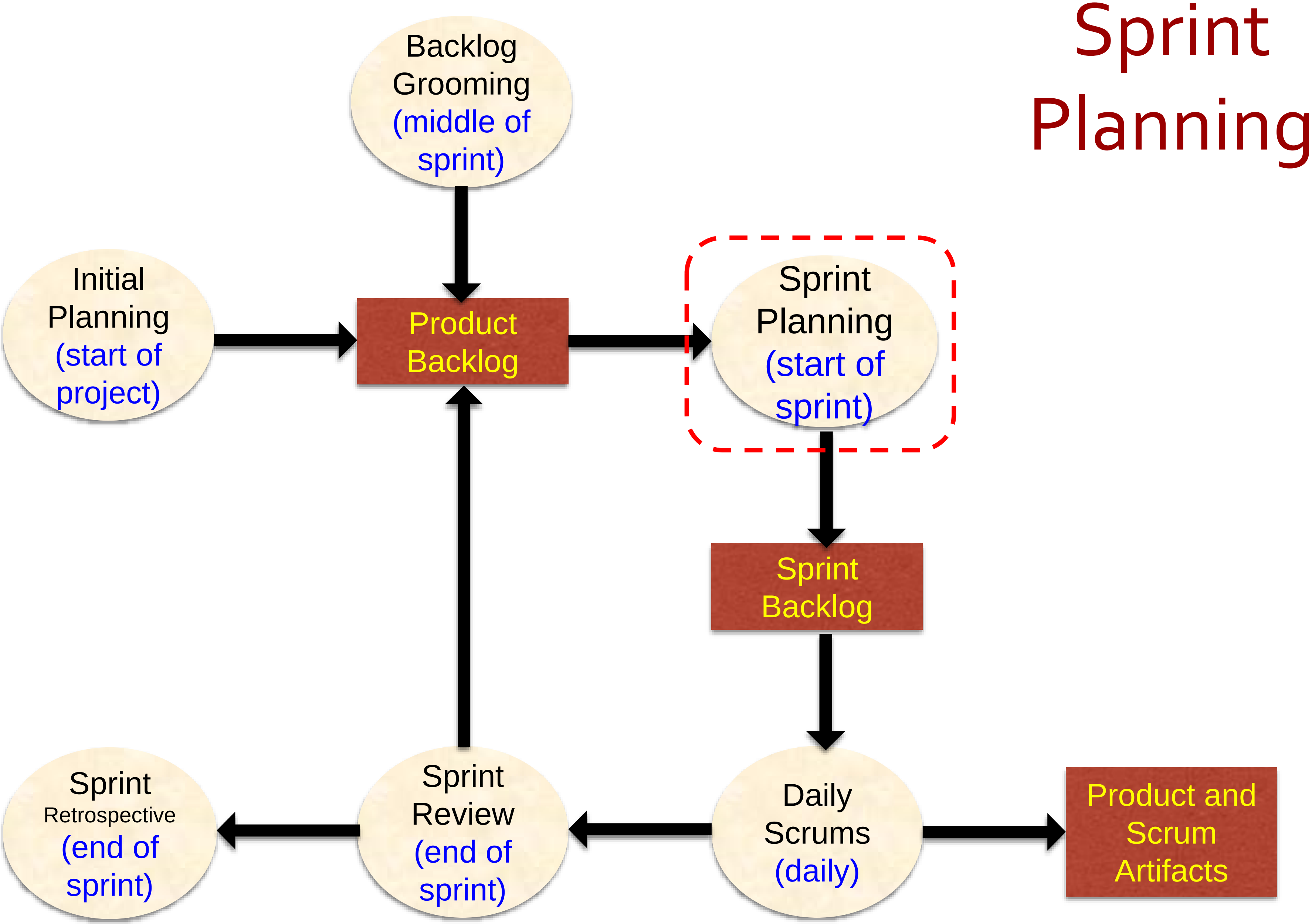
- The whole team meets during the Initial Planning meeting to:
  - Review/revise the initial User Stories
  - Review/revise/extend Acceptance Criteria for those stories
  - Estimate the initial User Stories
  - Rewrite high-priority epics (overly complex/vague User Stories) as detailed stories
    - Low-priority epics may remain unchanged in the Product Backlog until future meetings (e.g., Backlog Grooming, Sprint Planning)

Sprint Activities: Initial Planning Variations

- Scrum doesn't really dictate how to do Initial Planning ⇒ a lot of variation in industry
  - Some teams incorporate the initial planning activities into their first Sprint Planning Meeting
  - Others adapt their Backlog Grooming activity
  - Others may call it Release Planning

- CS471: entire team will meet with the sponsor (rather than Product Owner alone) and will generate/write the Product Backlog

Sprint Planning



Backlog Grooming (middle of sprint)

Initial Planning (start of project)

Product Backlog

Sprint Planning (start of sprint)

Sprint Backlog

Sprint Retrospective (end of sprint)

Sprint Review (end of sprint)

Daily Scrums (daily)

Product and Scrum Artifacts

Sprint Planning Meeting

- Determines what the team will do in the upcoming sprint

- Determines how the team will do that work

- Expect about 1 hour of sprint planning for each week of sprint

Sprint Planning Meeting – Part1: What to do?

- The first part of the meeting selects work for the upcoming sprint

- Product Owner decides what to implement in the upcoming sprint:
  - User Stories (with Acceptance Criteria)
  - Defect reports

- Epics (overly complex User Stories) can be rewritten if still necessary, but epics cannot be selected for current sprint. Why?

Sprint Planning Meeting – Part1: What to do?

- Product Owner reviews User Stories and Acceptance Criteria with the team
  - Update the US & AC as necessary

- Development Team decides if they can commit to do that work. Based on what?

Sprint Planning Meeting – Part1: What to do?

- Product Owner reviews User Stories and Acceptance Criteria with the team
  - Update the US & AC as necessary

- Development Team decides if they can commit to do that work based on the stories estimates/effort and historical throughput (i.e., velocity)

Sprint Planning Meeting – Part1: What to do?

- Product Owner reviews User Stories and Acceptance Criteria with the team
  - Update the US & AC as necessary

- Development Team decides if they can commit to do that work based on the stories estimates/effort and historical throughput (i.e., velocity)

- NB:
  - Product Owner decides what
  - Developers decide if

Sprint Planning Resolves Conflicts Between User Story Priority and Effort/Estimate

- Sprint Planning selects User Stories for implementation in the upcoming Sprint in order of their priority

- Sometimes, the next highest priority story won't fit in the sprint …

Sprint Planning Resolves Conflicts Between User Story Priority and Effort/Estimate

- Sprint Planning selects User Stories for implementation in the upcoming Sprint in order of their priority

- Sometimes, the next highest priority story won’t fit in the sprint because the team has already committed to higher priority stories

- Product Owner then chooses a lower-priority story that fits

Sprint Planning Resolves Conflicts Between User Story Priority and Effort/Estimate

- Remember: Product Owner prioritizes user stories
- Remember: Developers estimate the user stories
- Conflicts are resolved during the Sprint Planning meeting

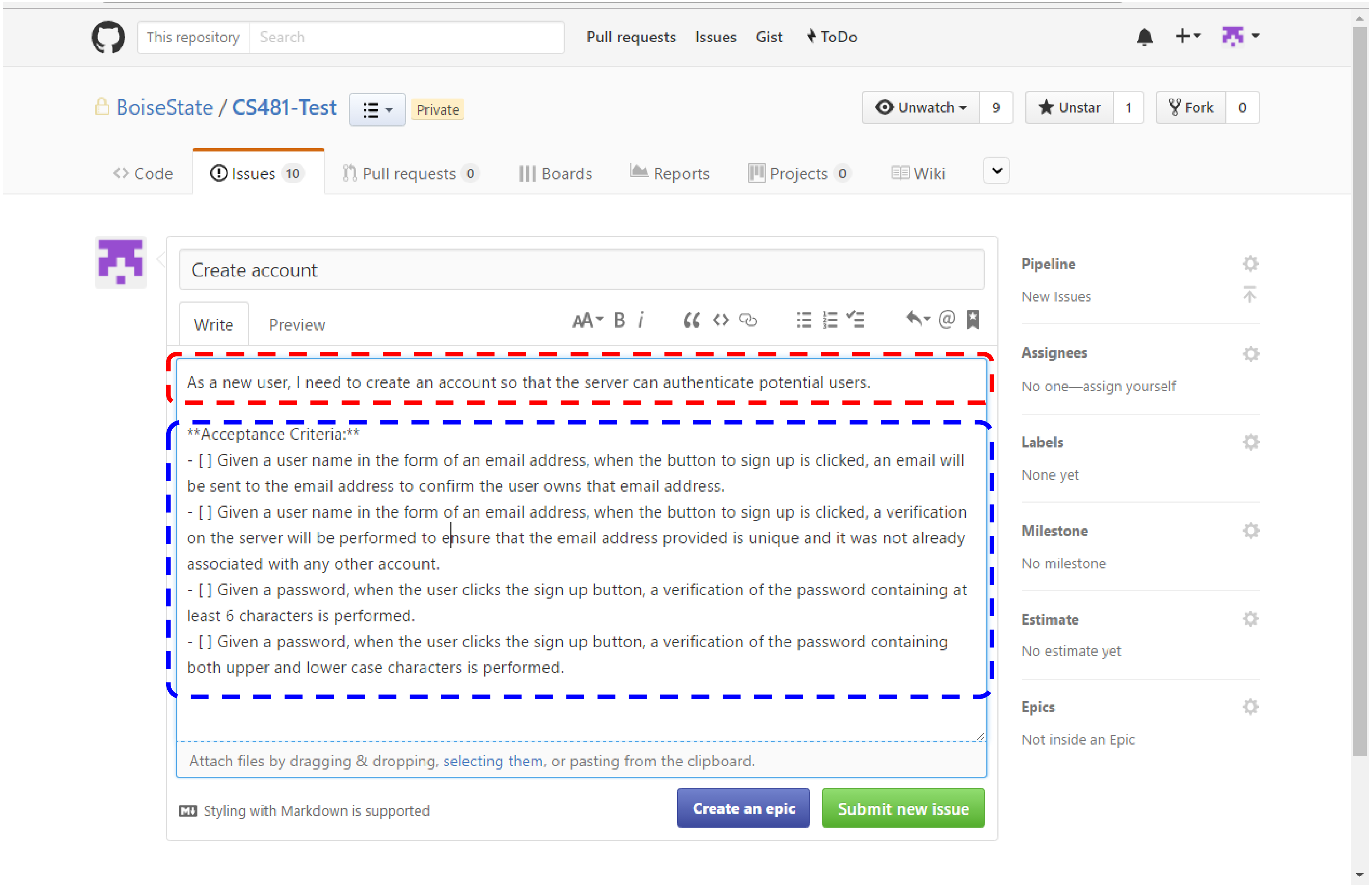
Example of old ways of keeping track of the Product Backlog in CS471: User Stories

Identifier	Priority	Description	Status	Estimate
1	A	As a new user, I need to create an account so that the server can authenticate potential users	ToDo	10
2	B	As an existing user, I need to login to access to my account	ToDo	5
3	B	As a logged-in user, I need to logout when I finish working with my account	ToDo	2
4	D	As a former user, I need to delete an account I am no longer using	ToDo	5

Example of old ways of keeping track of the Product Backlog in CS471: Acceptance Criteria

Identifier	Description	Status
1.1	Verify user name is a valid, user-owned eMail address	Pass
1.2	Verify password contains at least 6 characters	Pass
1.3	Verify user name is unique	Pass
1.4	Verify password contains both upper and lower case characters	Pass
	…	
2.1	Verify valid credentials provides user access to account	ToDo
2.2	Verify unknown user name is rejected	ToDo
2.3	Verify invalid password is rejected	ToDo
2.4	Verify session cookie keeps user logged-in if browser reopened	ToDo
2.5	Verify session cookie expires after 60 inactive minutes	ToDo
2.6	Verify session cookie will not expire during an active session	ToDo
	…	

Example Story + Acceptance Criteria in CS471/CS481 (GitHub+ZenHub)



Create account

As a new user, I need to create an account so that the server can authenticate potential users.

```
**Acceptance Criteria:**
- [ ] Given a user name in the form of an email address, when the button to sign up is clicked, an email will be sent to the email address to confirm the user owns that email address.
- [ ] Given a user name in the form of an email address, when the button to sign up is clicked, a verification on the server will be performed to ensure that the email address provided is unique and it was not already associated with any other account.
- [ ] Given a password, when the user clicks the sign up button, a verification of the password containing at least 6 characters is performed.
- [ ] Given a password, when the user clicks the sign up button, a verification of the password containing both upper and lower case characters is performed.
```

Sprint Planning Meeting – Part2: How to do it?

- The second part of the meeting decides how to do the work

- Developers decompose work (stories, defects…) into tasks
  - Tasks, unlike stories, are written in engineering language

Sprint Planning Meeting: Task Scheduling

- Developers volunteer to implement the Tasks and estimate them (based on their expertise)
  - Depending on team culture, developers may change the original estimate

- Project Managers/Scrum Master do not assign Tasks to Developers!

- Project Managers may encourage Developers to volunteer ☺

Sprint Planning Meeting: Sanity Check

- True or false?
- $estimate(UserStory_A) = \sum_{t\ is\ a\ task\ of\ UserStory_A} estimate(t)$

Sprint Planning Meeting: Sanity Check

- True or false?
- $estimate(UserStory_A) = \sum_{t\ is\ a\ task\ of\ UserStory_A} estimate(t)$

- Task Estimates are almost always more accurate

Sprint Planning Meeting: Sanity Check

- After all the Tasks are identified and estimated, check:
  - Does total planned work fit in the upcoming sprint’s time-box?

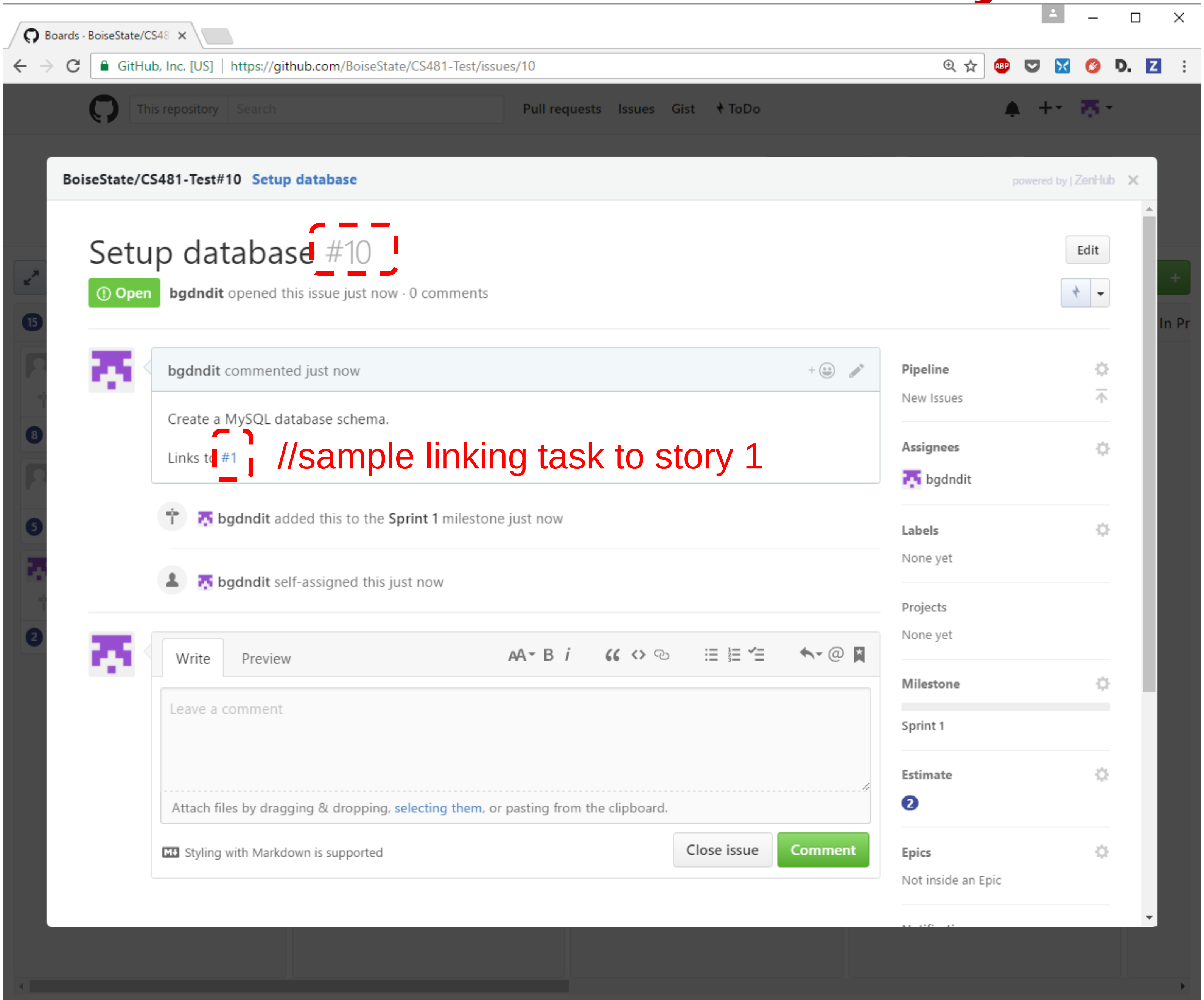
- If not, Developers re-negotiate stories (“what to do”) with Product Owner

- Resulting artifact is called the Sprint Backlog, a list of committed stories and tasks to complete in the next sprint

Example Sprint Backlog Tasks (Old Format)

Identifier	Story	Task	Owner	Estimate	Status
1.1	1	Create a new account screen	Todd	4	Pass
1.2	1	Verify eMail address is valid	Ann	2	Pass
1.3	1	Confirm eMail address	Ann	8	Pass
1.4	1	Enforce the password policy	Jason	4	Pass
1.5	1	Create database table for user accounts	Arti	2	Pass
1.6	1	Create new account entry in database	Arti	4	Pass
4.1	4	Create delete account screen	Todd	2	InProgress
4.2	4	Delete account and logoff user			ToDo

Task #10 linked to Story #1



BoiseState/CS481-Test#10 **Setup database**

Setup database #10

Open **bgdndit** opened this issue just now · 0 comments

bgdndit commented just now

Create a MySQL database schema.

Links to #1

//sample linking task to story 1

bgdndit added this to the **Sprint 1** milestone just now

bgdndit self-assigned this just now

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Pipeline
New Issues

Assignees
bgdndit

Labels
None yet

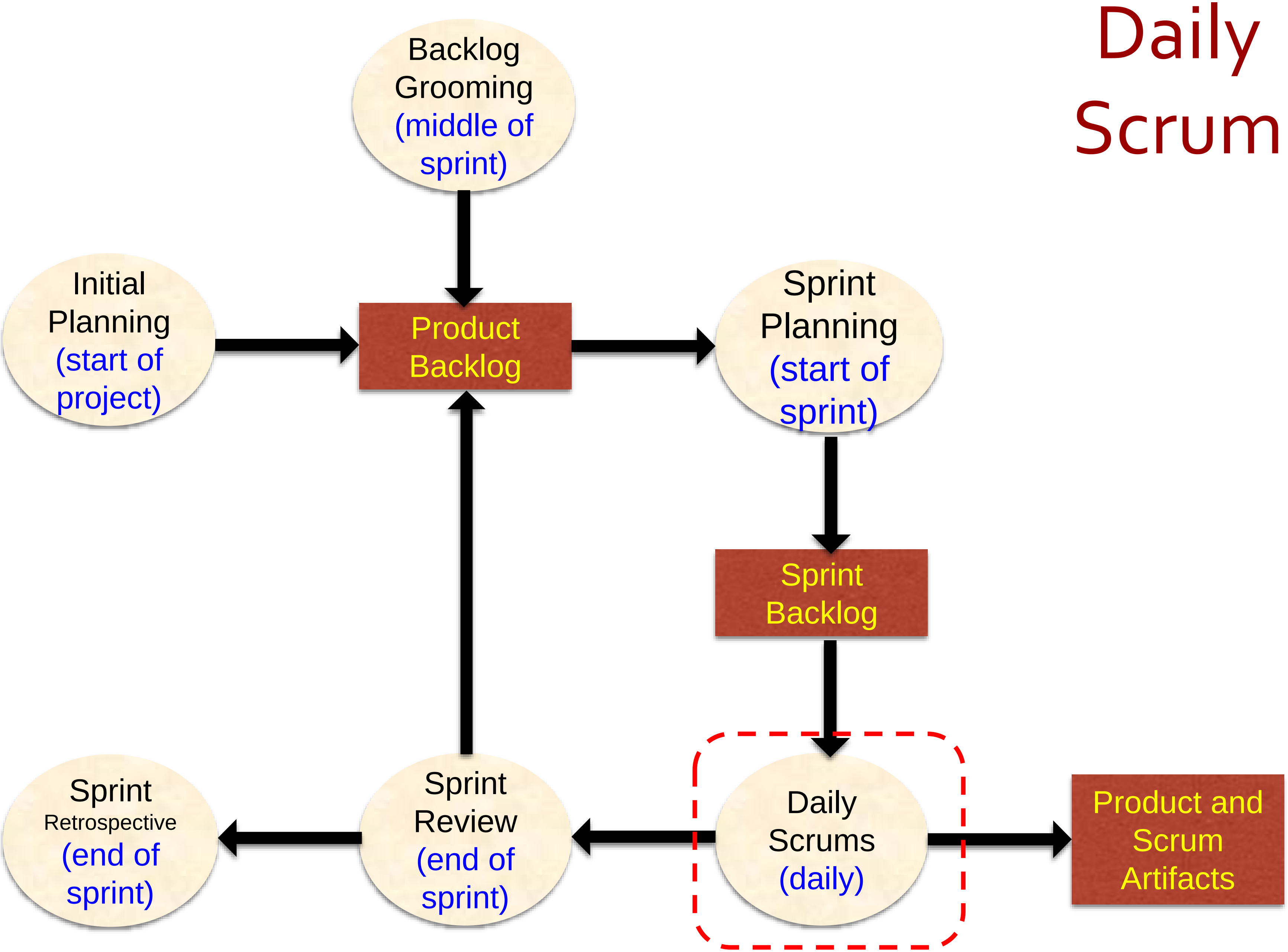
Projects
None yet

Milestone
Sprint 1

Estimate
2

Epics
Not inside an Epic

Daily Scrum


Backlog Grooming (middle of sprint)
Initial Planning (start of project)
Product Backlog
Sprint Planning (start of sprint)
Sprint Backlog
Sprint Retrospective (end of sprint)
Sprint Review (end of sprint)
Daily Scrums (daily)
Product and Scrum Artifacts