12 - Teamwork

FINTECH 512
Software Engineering

Duke

Collaboration

"The interaction of two programmers looking over a program that either one of them could have worked out is entirely different from the interaction of two programmers working on separate parts of a whole which is too great for either one to produce.

The difference lies in the way conflicting demands are resolved. In the first case, resolution of conflict is the thinking process of one person—aided perhaps by other people, but always under that one person's control. **In the second case, conflicting technical demands are translated into potential interpersonal conflicts, and a social mechanism must be formed to resolve them**.

Weinberg, Gerald. The Psychology of Computer Programming: Silver Anniversary edition

Duke

Group projects, good and bad

- What are your worst experiences with group projects?
- When is it bad to work together on something?
- What makes it difficult to work with another person?

- What are your best experiences with group projects?
- When is it good to work together on something?
- What makes it easy to work with another person?

Duke

Teamwork requires…

- A compelling goal
- Complementary skills
- Interdependent work
- Mutual accountability

"Creating an Environment for Teams", Esther Derby, www.estherderby.com

Duke

Teamwork: A compelling goal

- "True teamwork is a response to a compelling goal or challenging problem that has meaning for the team's customer and the team members themselves"

- Three questions:
  - What problem are we trying to solve?
  - What benefit are we trying to create?
  - Who really need this?

- Your goal: Build a BigBucks site that meets our requirements so you are prepared to do similar work when needed.

- My goal: Students learn and apply skills that will help them on a software development team and in interviews by seeing a small project through to completion.

Duke

Teamwork: Complementary skills

- On this, and most, projects, there are a variety of tasks that people may specialize in to produce a coherent product:
  - Java design and coding
  - Database design and management
  - User Interface design and construction
  - Operations; software installation, monitoring, and updating
  - Project management
- We address each of these in 512
- You should have some familiarity with all of them
- You may wish to focus on one in particular
- Successfully coordinating with your team to cover all aspects of the project makes the project
  A. Possible
  B. Easier

"Creating an Environment for Teams", Esther Derby, www.estherderby.com

Duke

Teamwork: Interdependent work

- Clinic example:
  - Five doctors, five nurses
  - Doctors responsible to patients but not necessarily each other
  - Nurses responsible to patients but not necessarily each other
  - Doctors and nurses may be interdependent

"Creating an Environment for Teams", Esther Derby, www.estherderby.com

Duke

Teamwork: Mutual Accountability

- Accountability

Observation of results: did you do the thing? And did you do it well?

- Consequences:

What happens if you did not do the thing?

- Balancing
  - On One Hand: Helping the team out
    - If the work is not getting done and you can do it, great
    - It's important to work together and help each other out
  - On the Other Hand: if the message becomes, "Someone else will do the work"
    - Sets up a bad incentive structure
    - What if happens repeatedly with the same person?
  - How do you balance?
    - Helping out is great provided the work balances out
    - If someone is not pulling his/her/their weight, need to fix the situation

"Creating an Environment for Teams", Esther Derby, www.estherderby.com

Duke

Consequences

- In class
  - Peer pressure
  - Grades
- In industry
  - Peer pressure
  - Lack of raises
  - Layoffs, firings
  - Companies go out of business

Duke

Consequences

- If you can’t resolve an issue in your group
  - Document the problem, and include your desired course of action
  - May not agree on all aspects: include description of disagreements
  - Send documentation to your professor
  - Follow up with outcomes (e.g., did the person complete? Not complete?)
  - Professors will determine if issues will affect individual grades

Duke

For the semester, accountability comes in the form of grading… look at project grading

- "Consistency"… team and individual effort over the course of the project, as measured by accomplishing project requirements as recorded in your issues in the project repository.

Date	Method1	Method2	Method3
3/4/22	1	1	1
3/11/22	15	5	1
3/18/22	15	5	1
3/25/22	15	5	1
4/1/22	20	5	1
4/8/22	20	30	49
4/15/22	14	49	51
Max Grade	100	100	80

Duke

Teamwork: Support for the team

- Context for the work, an understanding of how it fits into the big picture
- Material support; budget, space, tools, …
- Access to expertise not available on the team
- Feedback loops; connection to the customer and to the organization

Duke

Scrum

- "Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems."

https://scrumguides.org/scrum-guide.html

Duke

Scrum

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

- A Product Owner orders the work for a complex problem into a Product Backlog.
- The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
- The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
- *Repeat*

https://scrumguides.org/scrum-guide.html

Duke

Scrum for 512

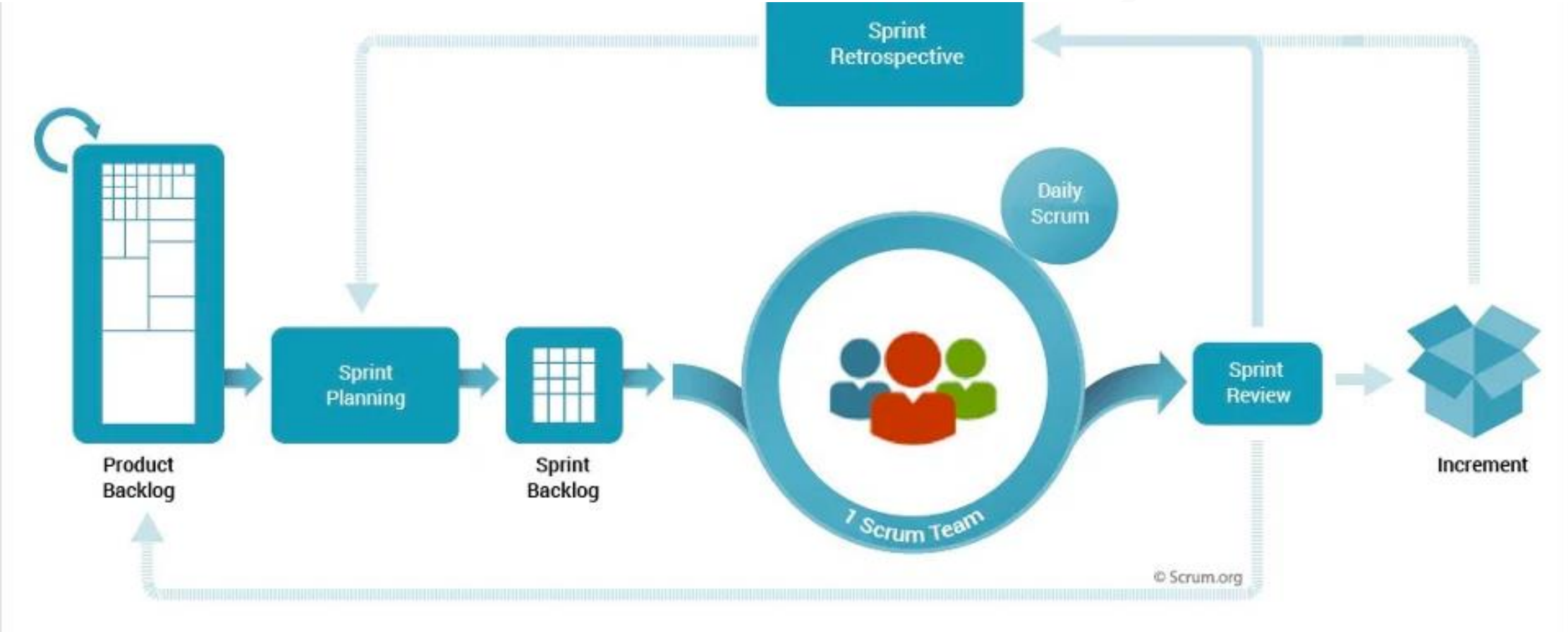
In a nutshell, Scrum requires a Scrum Master (TA) to foster an environment where:

- A Product Owner (each team) orders the work for a complex problem into a Product Backlog.
- The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
- The Scrum Team and its stakeholders (TA, instructors) inspect the results and adjust for the next Sprint.
- *Repeat*

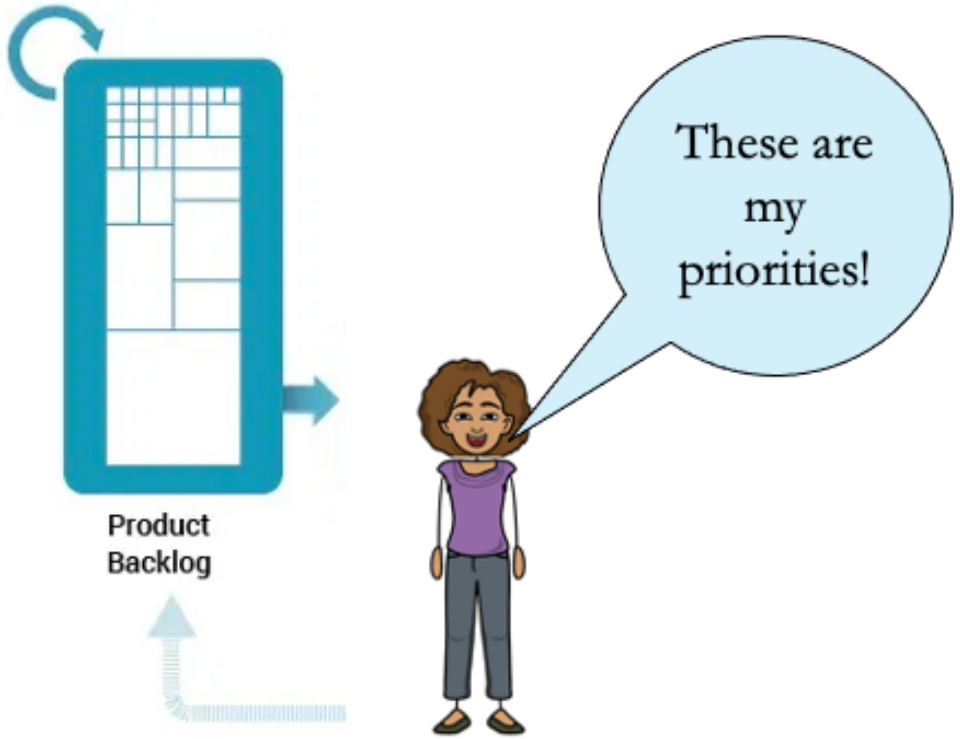
https://scrumguides.org/scrum-guide.html

Duke

Scrum Process Overview



Duke

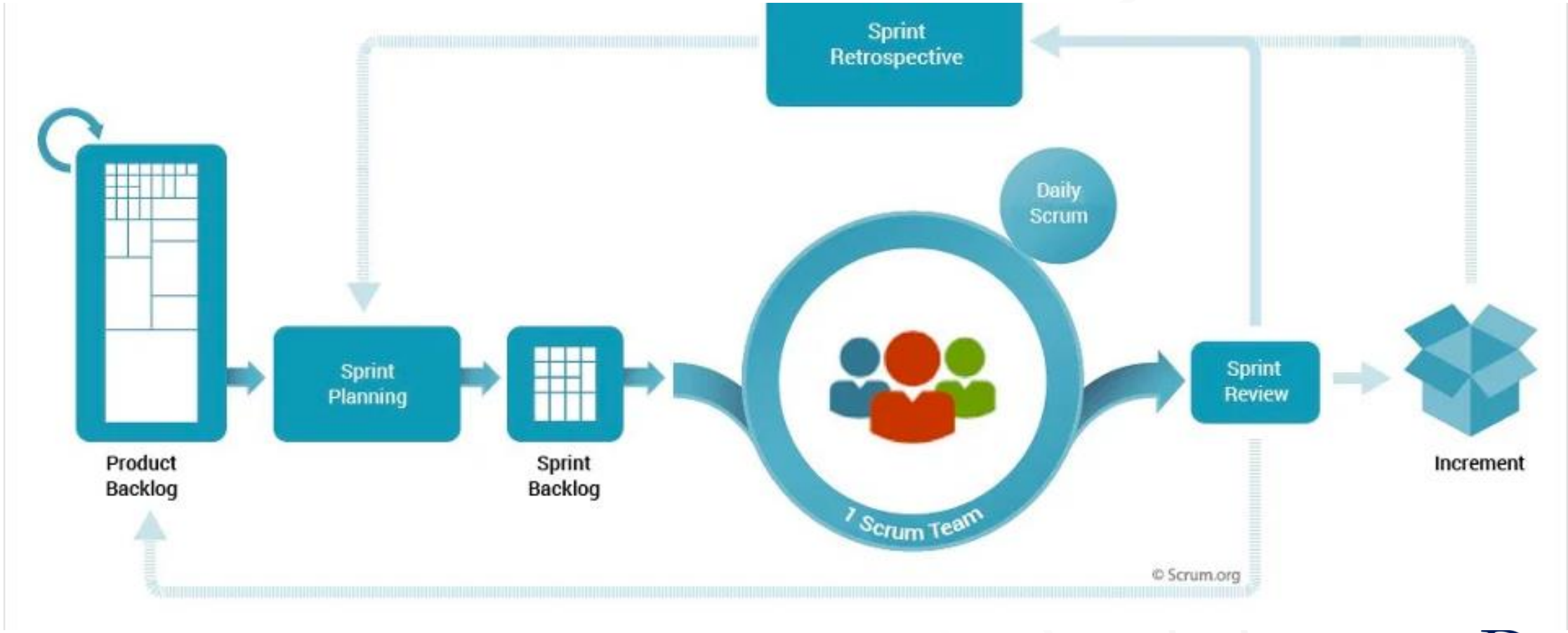


Product Backlog

1. Laser sharks guarding castle
2. New minion training facility
3. Fresh baked cookies daily in lair
4. ….

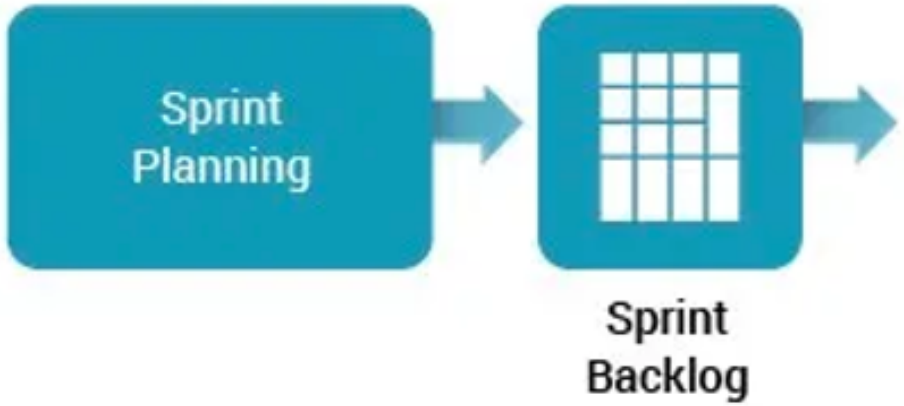
Duke

Sprint Planning: Select features for next sprint, break down into tasks



Duke

Note: Some tasks might not fit in current Sprint Backlog. Work with product owner to make decisions & select tasks that match spring duration



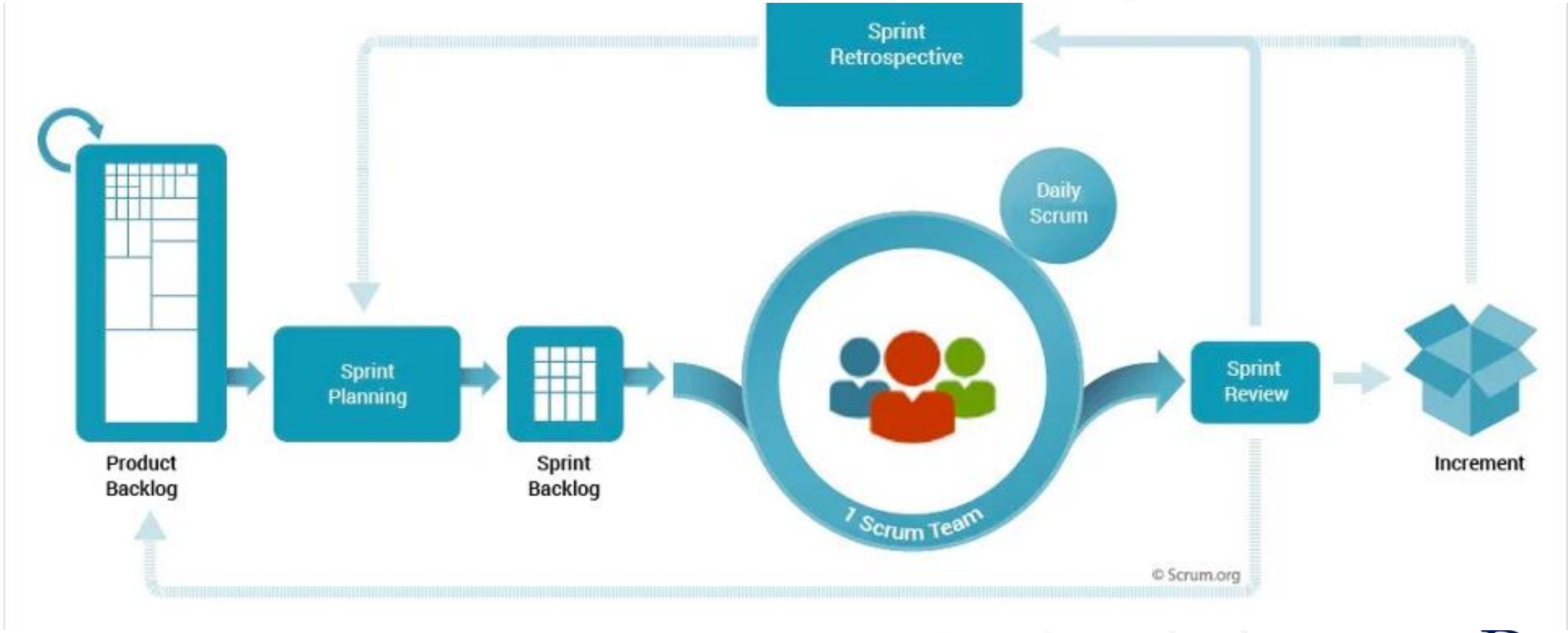
Sprint Backlog

1. **Laser sharks guiding castle**
   - Build head mount lasers (8 hours)
   - Fill moat (1 hour)
   - Train group 1 sharks (6 hours)
   - Train group 2 sharks (6 hours)
   - Train group 3 sharks (6 hours)
   - Deploy sharks to moat (0.5 hours)
2. ~~**New minion training facility**~~
3. **Fresh baked cookies in lair**
   - Interview cookie chef (5 hours)
   - Clean lair kitchen (2 hours)

[redacted]

Duke

Sprint: Development work: pull tasks from sprint backlog, build, test, review



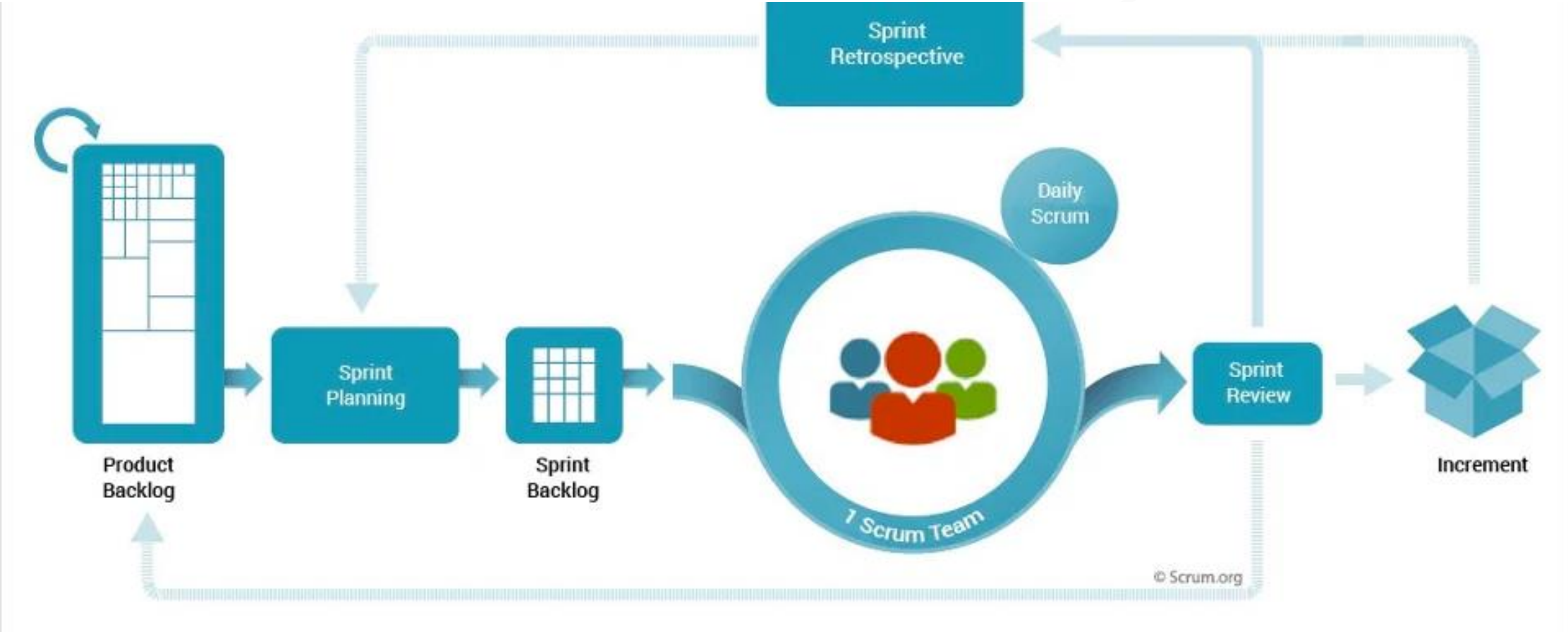
Duke

Daily Scrum

- Classic questions:
  - What have you been working on?
  - What will you work on next?
  - Do you have any blockers?

Duke

Sprint Review: Demonstrate results, ask for feedback



Duke

Teamwork: First meeting

- Set expectations up front
  - What are your standards?
  - What is your plan for a schedule?
  - What accountability mechanisms do you have?
- Expectation setting should be a full team activity
  - Everyone agrees
  - Not one person dictating
- Concretely for this class:
  - Discuss your goals/schedule/etc… align
- Communicate problems proactively
  - “I can’t figure this out”
  - “Something came up, I’m behind”
  - “This doesn’t work the way we expected”
  - “I finished this sooner than I expected, anything else I can do?”

Duke

Teamwork: First Sprint (from project document)

- Watch and discuss ’How to Design a Good API and Why it Matters’
- Read through the project requirements and
  - Decide in what order you will work on the numbered project requirements
  - Decide which requirement(s) to begin with and who will work on each requirement
  - Create (per the PSP script) the planning issue for the requirement(s) you will work on
- Write a Project Status Report

Duke

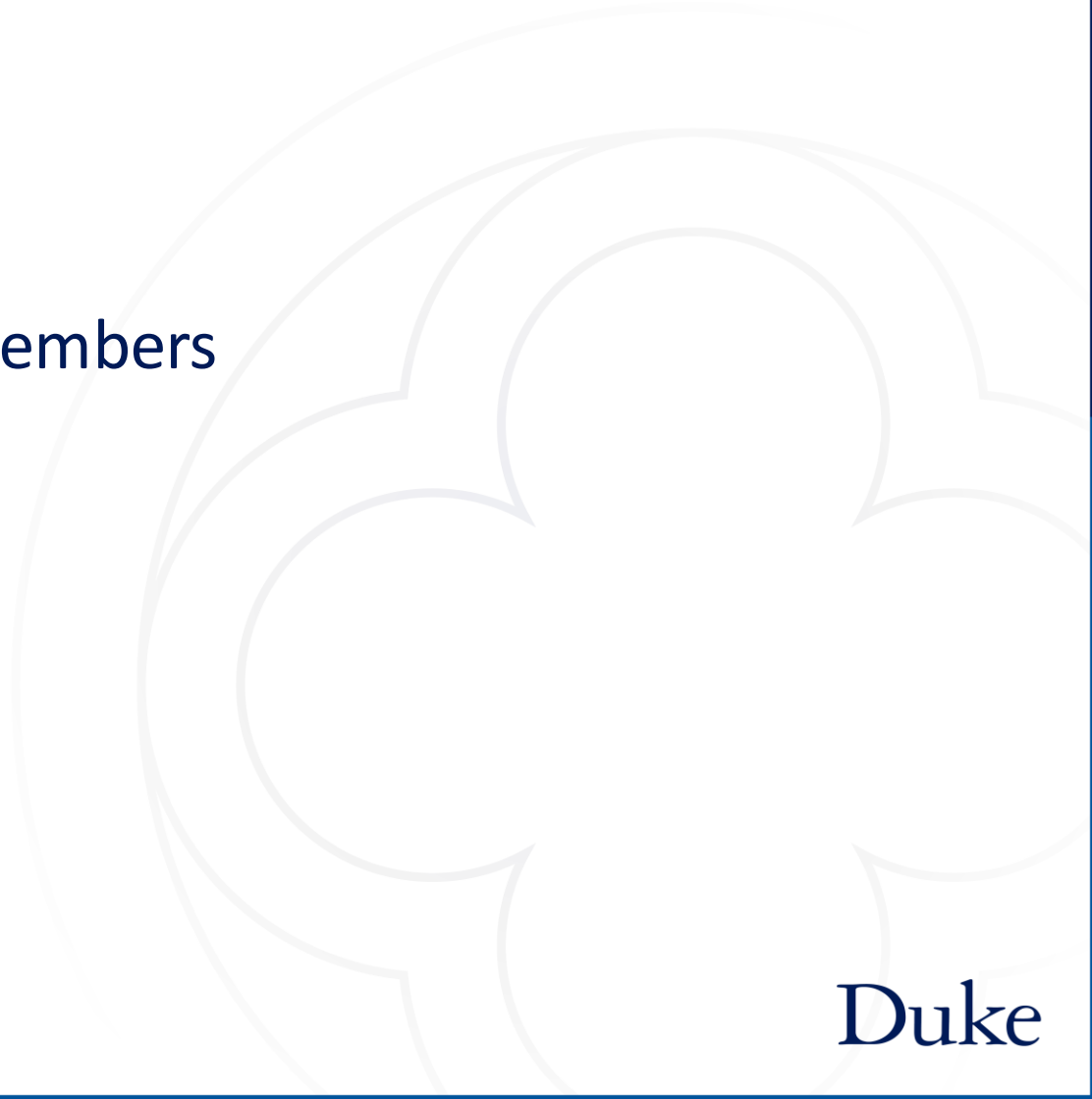
Teamwork: Weekly Sprint Meetings with TA

- Review your progress on the plans and requirements you've worked on so far
- Where possible, demonstrate completion of requirements, including all 'paperwork' (closed issues for planning, review, testing)
- Read through the remaining project requirements and
  - Decide in what order you will work on the numbered project requirements
  - Decide which requirement(s) to begin with and who will work on each requirement
  - Create (per the PSP script) the planning issue for the requirement(s) you will work on
- Write a Project Status Report

Duke

Teamwork: Technical aspects

- Git workflow
- Code Review
- Integrating code from multiple team members



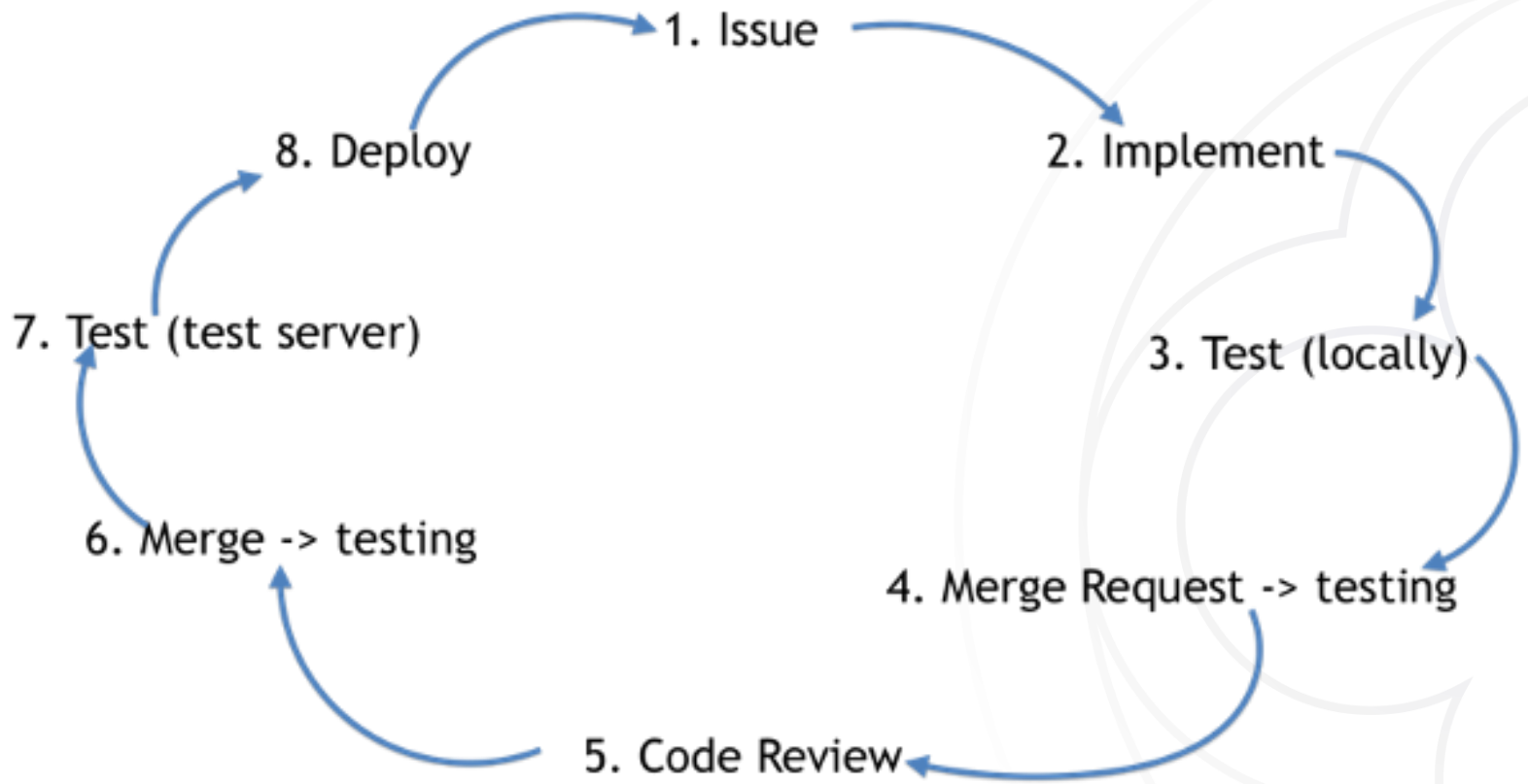
Duke

Git for Teams

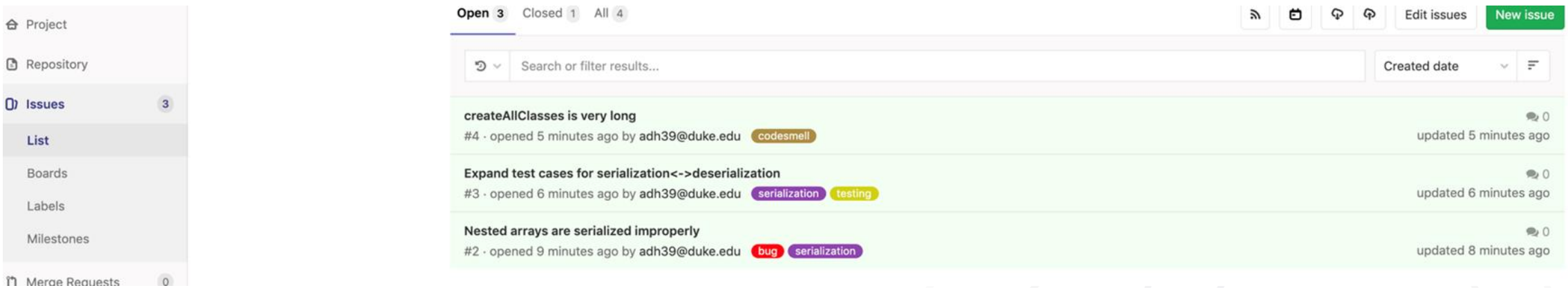
- So far: individual use of git, primarily on the repo's master branch
- For the project:
  - Create separate branches for each new feature (or bug)
  - May be several different features/tasks being developed at once
- Common scheme:
  - Master branch holds stable code, suitable for release, putting into production
  - 'Staging' branch holds 'almost ready' code, undergoing tests and preparation for release
  - Development branches, often organized by feature, created by single developer for working on their part of the sprint

Duke

Life Cycle of Feature Development



Duke



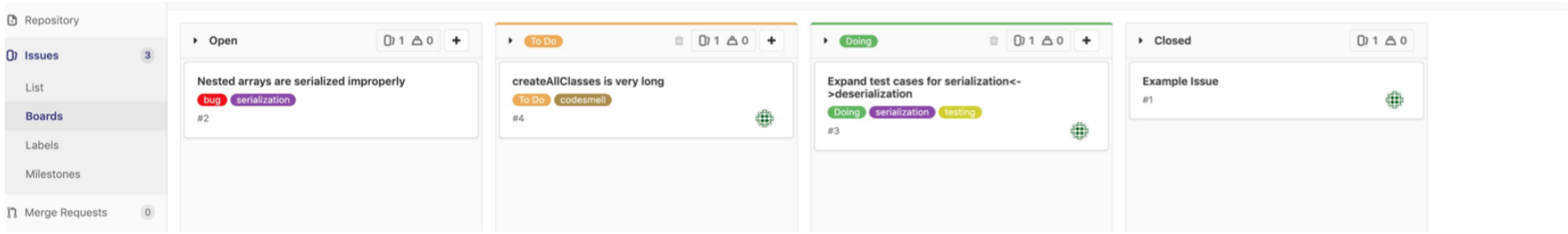
Issues

Track list of open issues

- Bugs
- Features
- Improvement

Assigning what needs to be done with time estimates

Duke



Gitlab "Board" | HELPS USERS SEE STATUS

Duke

Implementation: Step 1

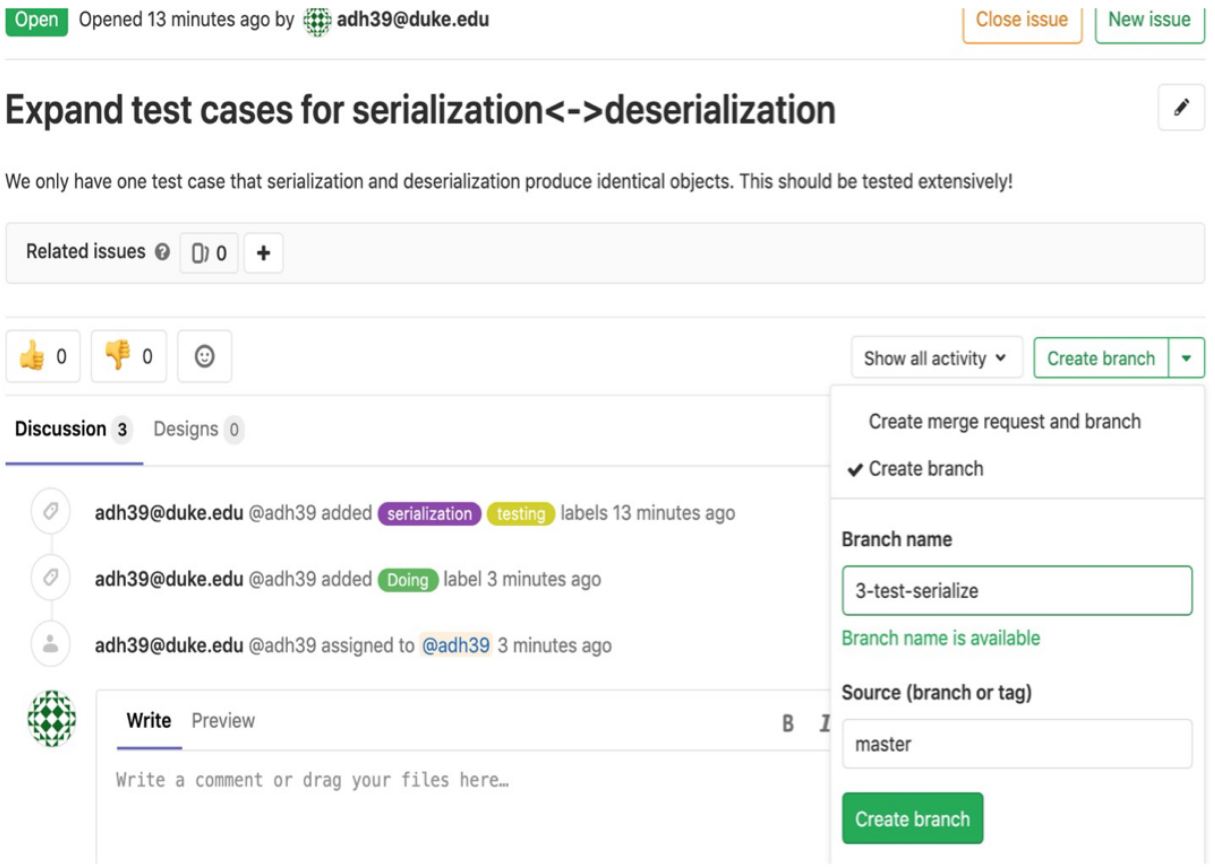
Command line:

"git branch"

OR

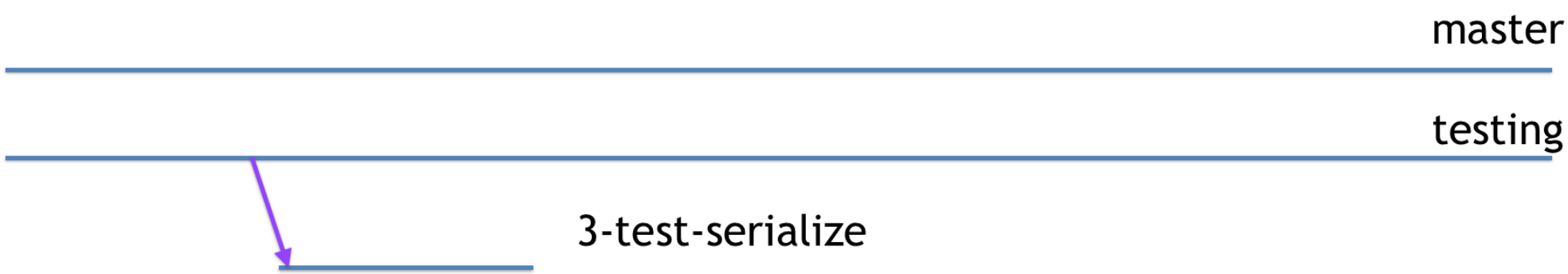
GitLab UI

(ties directly to issue)



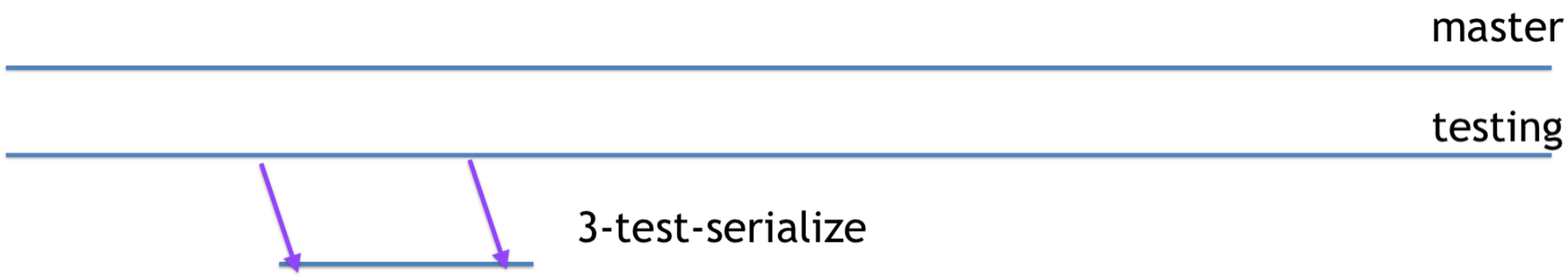
Duke

Feature Branch for an Issue



Feature branch: a developer's own branch for a particular issue

Feature Branch for an Issue



If anything changed while you were working on this isse

- Merge those changes in (git merge testing)
- Test again!
- Repeat until convergence..

Duke

Merge (Pull) Request

Instead of pushing to testing (or master), we ask for review

- Naija wrote the code
- Anna will review before accepting

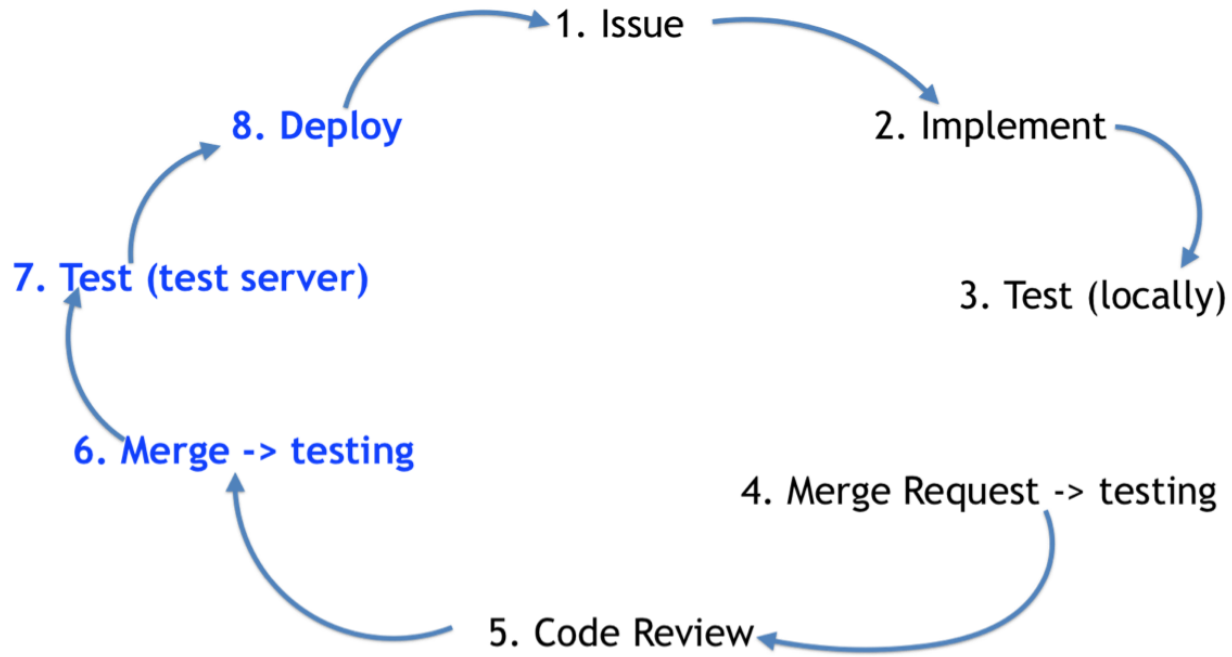
Duke

Code Review

Anna will check Naija's code quality

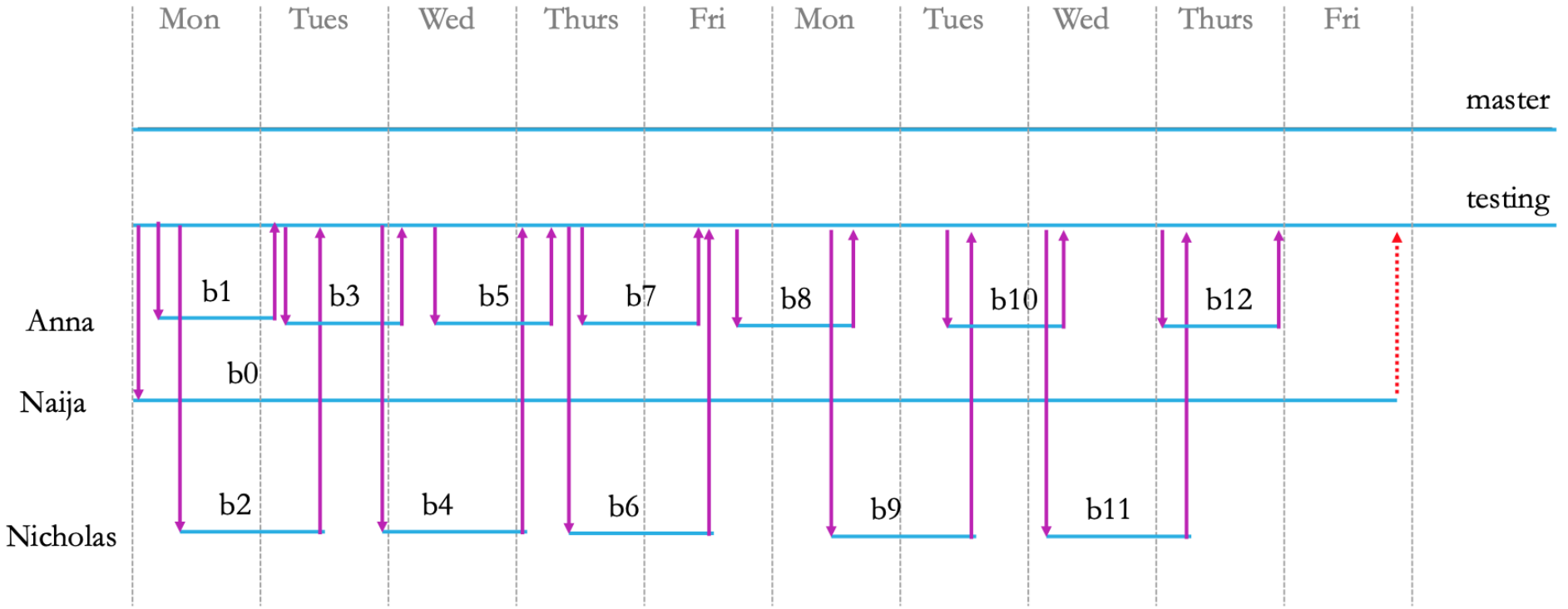
- Does the code meet the organization's coding standards?
  - Formatting, naming, commenting, etc?
- Are the test cases sufficient for the change?
- Are there any apparent issues (e.g., security, missing cases, …)?
- Are there any code smells or design issues that should be resolved?

Life Cycle of Feature Development



Duke

Naija checks out b0 and work on it for 2 weeks.
What might be the problem when she's ready to check the code back it?



Anna and Nicholas wrote a lot of other code in this time -> much has changed -> many merge conflicts ☹

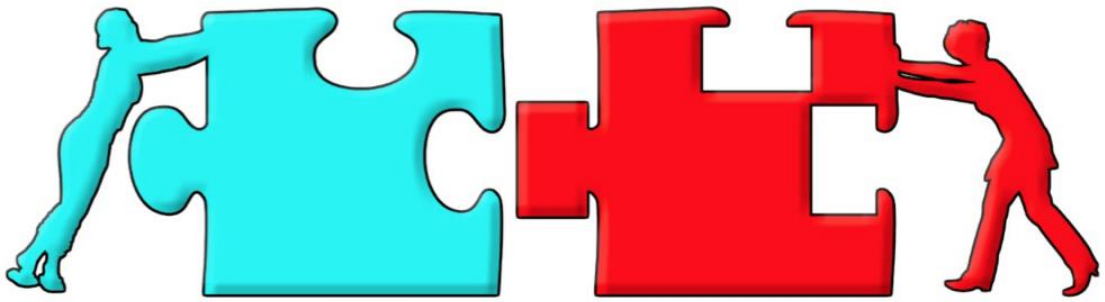
Duke

Code Divergence

Naija's code: is based on what was happening 2 weeks ago, but many things have changed

She has a large (messy/nasty) merge to resolve

Her code might be hard to integrate



Remember this?

Duke

Curing the Problem

Doctor, doctor!

It hurts when I go a long time without integrating my code with the main line of development!

What will she say?

Source: http://theconversation.com/whats-in-a-title-when-it-comes-to-doctor-more-than-you-might-think-127979

Duke

Rule of Thumb

Integrate AT LEAST daily

If the feature is too large, then it's probably not broken down enough

Duke

Next Time: Continuous Integration

Duke