

Intro - review [Team Management Plan](#) sections - we'll work on sections V and VI -

V. *Systems to ensure that critical tasks are completed on time*

VI. *Procedures for documenting, cataloging, and archiving information*

NOTE: draft of team management plan due next Thursday, so I can review and offer feedback before the final draft is due the following week.

Set up GitHub organization

- Create organization, name it something like your capstone name
- Invite collaborators (change permissions if required)

Sample setup

- Each team member (in parallel) creates a repo related to some aspect of the dinner party – bread, salad, lasagne, side dish
 - These we can delete later
- Everyone go into the bread repo, check that everyone is “watching”

Issues

- The person who created that repo, find a recipe for bread online somewhere, start a new issue asking for feedback on the recipe
 - Everyone else respond to the issue, perhaps taking turns, perhaps suggesting new recipes, perhaps giving thumbs ups etc. Note – you should be seeing these issues in your e-mail – and you can reply in Github, or via e-mail!
- Note that Issues can be questions, problems, ideas, tasks, or just information.
- Everyone create a new issue related to bread – the steps of the recipe
 - Mention people to issues (@)
 - Link issue to other issues (#)
 - Create task lists inside the issue (button, or - [])
 - Assign labels to issues
 - Note on the issues page all the ways to filter!
- Create two milestones related to bread (e.g., bread is ready to bake, dinner time), and users assign issue to those milestone if it makes sense (e.g., maybe “dinner time” only assigned to the FINAL bread task? Decide as a group)
- Note the filters: field – you can type things in up here too – search, filter, etc
- Link to best practices: <https://rewind.com/blog/best-practices-for-using-github-issues/>

Project - organization level

- Now: go up to the organization level, create a new project – call it something like “capstone dinner”
- In View 1 (table mode)
 - create items (as drafts) for the non-bread parts of dinner (incl dessert – everyone creates an item)
 - create items (from issues) - select issues from the bread repo – create a task from an issue!
 - Explore the other columns – assignees, status – other columns potentially as well
 - Each person assign their task to someone else in the group.

- One person change the status of their item
- In View 2 (board mode)
 - Note that you can create items here as well, just like in "table" mode
 - arrange the tasks as makes sense... we'll do the bread first, everything else is to do or not yet assigned.
 - As a group, talk through the bread baking process and move tasks accordingly – most get moved to "done" but don't yet move the final "baking" task to "done"
- In either view
 - take one of the draft items (e.g., lasagne), convert to an issue in the appropriate repository
 - Go look in that repository, verify that there's new issue - can edit it to flesh it out!
 - Add this issue to the "dinner time" milestone

Automation across items and such

- go back to the repos, choose the bread repo, click on "issues" – examine the "milestones" – even though things moved to "done", still 0% complete? Must close the issues!
- choose the first one or two bread-related issues/items, and write a message then close the issue
 - note that the GitHub bot automatically changes the status of this item to "done"
 - verify back on the "projects" page.
- With pull requests, can close pull requests automatically with issues - maybe next week?

More on repositories:

<https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories>

Discuss communications - Use the proper tool for the task at hand!

- Communications with external people? Probably e-mail
- Internal comms for quick questions/answers, probably Slack
- Internal comms related to tasks - incl questions/answers, assigning, progress, completing - probably Github Issues
- YANSS episode talks about some of these ideas and more -
 - [YANSS 251 – Why Nick Sonnenberg doesn't believe there just aren't enough hours in the day to get everything done](#)

Now for student projects!

- go back to organization – create a new project (optional, delete sample project)
- create some repositories related to the various pieces of your overall capstone (e.g., data prep, analysis 1, analysis 2, R package, Shiny app) (optional, delete sample repositories)

- start to populate the project with items representing the major steps from your Gantt chart.
 - For the most immediate steps, flesh out with a few issues – assign issues, assign status, etc.
- Create some milestones (important events e.g. reviews, presentations, etc) and assign tasks to those milestones, etc.
- Discuss how these things can be used in your group - start to consider which tools for which purposes, how to use repos, issues, and projects effectively - perhaps start drafting some text for parts V and VI of team management plan
 - some overlap with roles and responsibilities
 - Cross tie to documentation plan, testing plan, timeline and milestones, data management plan (part of solution design)