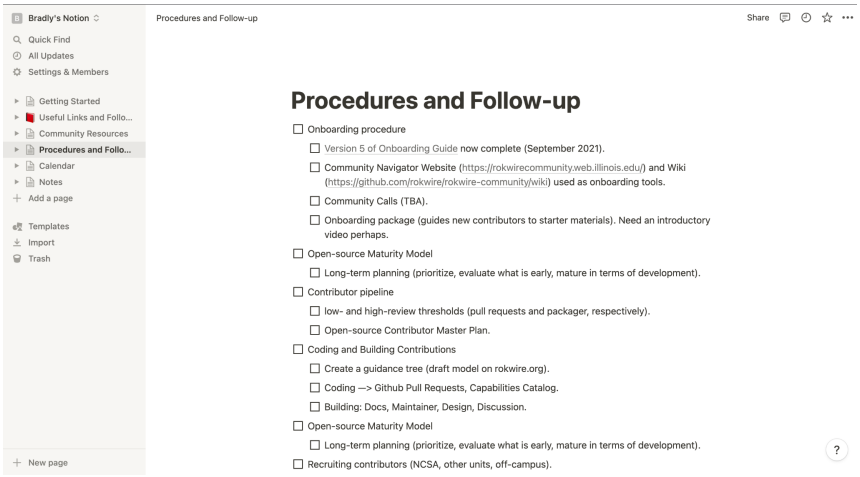
Collaboration Tools I

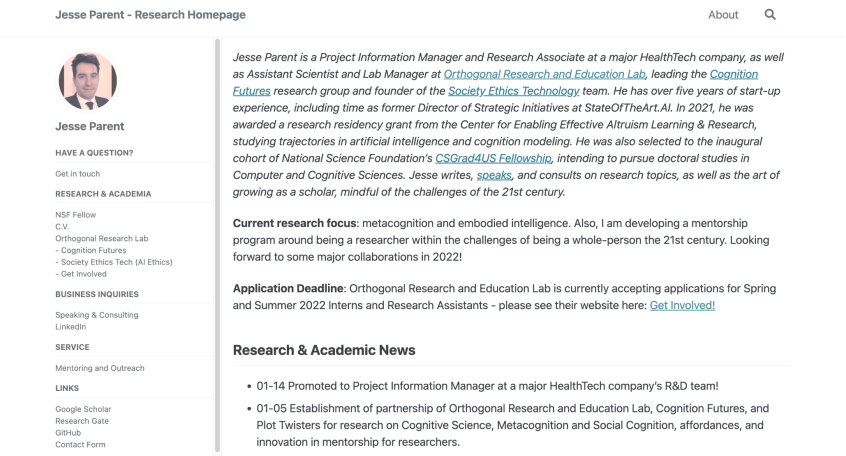
What are Collaboration Tools?
Collaboration tools allow for better ways to organize individual work, seamless work between contributors, and a way to easily onboard new contributors into an existing workflow.

Notion: https://www.notion.so/

Notion is a good tool for building checklists, tables, and outlines for organizing resources and media assets. Notion pages can be used to create web documents, or shared amongst collaborators.



Jesse Parent's website (built in Notion): https://jesparent.github.io/

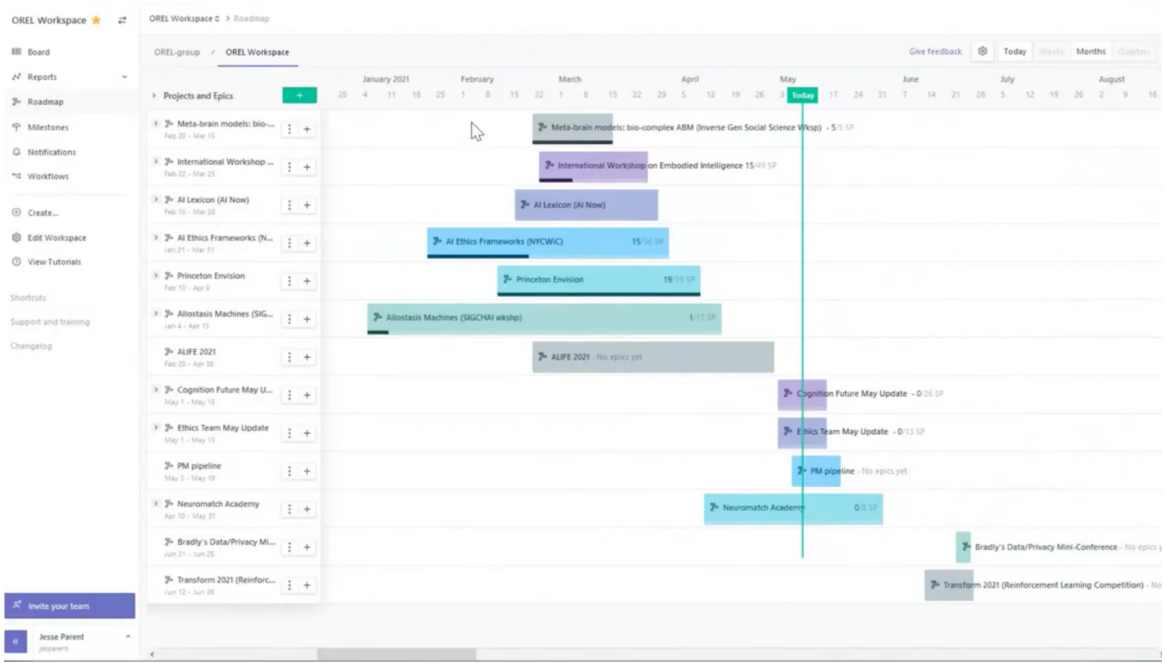


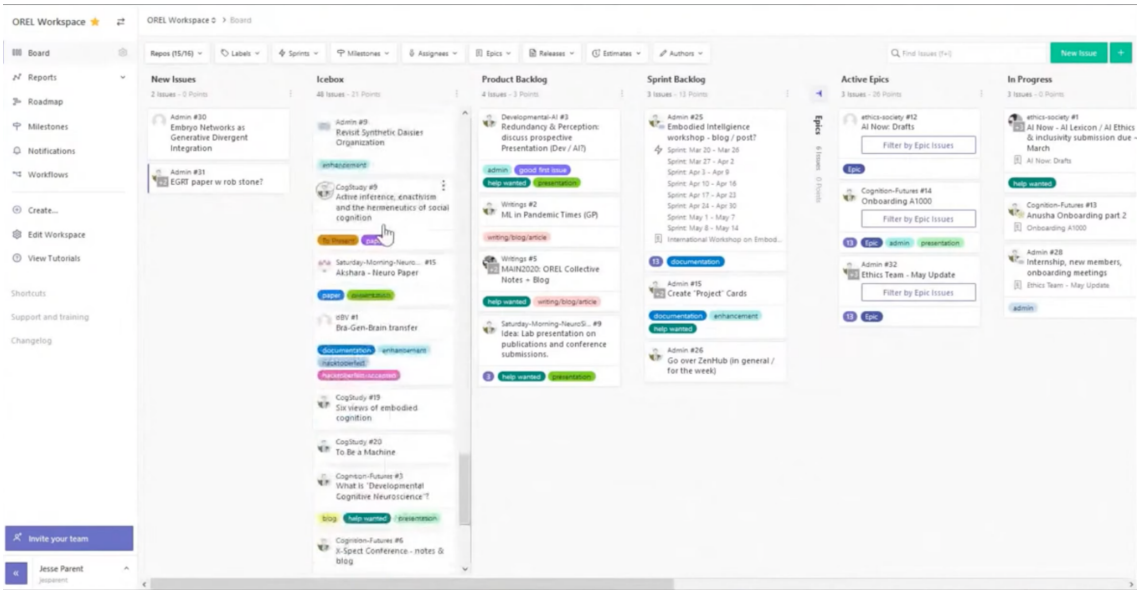
Obsidian: https://obsidian.md/

Obsidian consists of vaults where you can store and organize notes around themes. Here is an example of a student (Amanda Nelson) organizing a set of concepts using Obsidian (see this video clip for more).

Zenhub: https://www.zenhub.com/

Zenhub is a tool that allows you to organize milestones from a set of Github issues. There is both a timeline view and a Kanban board view, which can be helpful for seeing the big picture.





Organizing tasks using a timeline view allows us to track milestones, which are essential for complex tasks like public events.

Milestones:

As the ZenHub demo shows, milestones are specific points on a project timeline. They may act as anchors or as concrete goals with an associated date. Prospective milestones can be flexible, depending on how much is achieved. Providing a concrete goal is good motivation for focusing efforts, particularly when open source contributions are involved.

Milestones also define points at which other team members might give input. Our milestones might be deliverable dates or dates when other dependencies are expected to be available. In this case, it is important to be flexible.

- A good tip for timeline building is to have parallel activities going on. If a dependency delays you or otherwise ties up your resources, you can focus on another task or milestone.

- A recommended read on Agile management processes and the rise of software development: https://logicmag.io/clouds/agile-and-the-long-crisis-of-software/

Epics:

ZenHub also uses a type of organization called “epics”, which encapsulates a theme of work. Epics are similar to Github issues in that they are both organized by subject. However, epics introduce dependencies to your Github issues, and they can be used in tandem. This is different from sprints, where things are related by time. In sprints (as we covered previously), tasks are accomplished as an accelerated workflow. Using sprints, Github issues, and Zenhub epics together can be a very powerful way to synchronize community contributions and reduce conflicts with a top-down software development cycle (imposed by a core team of internal developers).

- An Introduction to Zenhub Epics: https://help.zenhub.com/support/solutions/articles/ 43000010341-an-intro-to-zenhub-epics

- Working with Epics in Github: https://blog.zenhub.com/working-with-epics-in-github/

Some readings:

What I’ve learned about open source project management: milestones (Tom McFarlin blog): https://tommcfarlin.com/open-source-project-management-milestones/

Milestones (GitLab docs):
https://docs.gitlab.com/ee/user/project/milestones/

The role of Milestones in Agile project management:
https://blog.zenhub.com/what-is-a-milestone-in-agile-project-management/

Sprints vs. Milestones:
https://medium.com/@confeurhq/sprints-vs-milestones-6fe700d101f9

A set of completed milestones from the Open Source Design project:
https://opensourcedesign.net/milestones

Other Project Management Approaches:

My favorite open project management tools:
https://opensource.com/article/21/3/open-source-project-management

Defining a Minimal Viable Community (Commsor)
https://www.commsor.com/post/minimum-viable-community