Collaboration on GitHub

QBS 101.5: Applied Data Science

Simon Stone

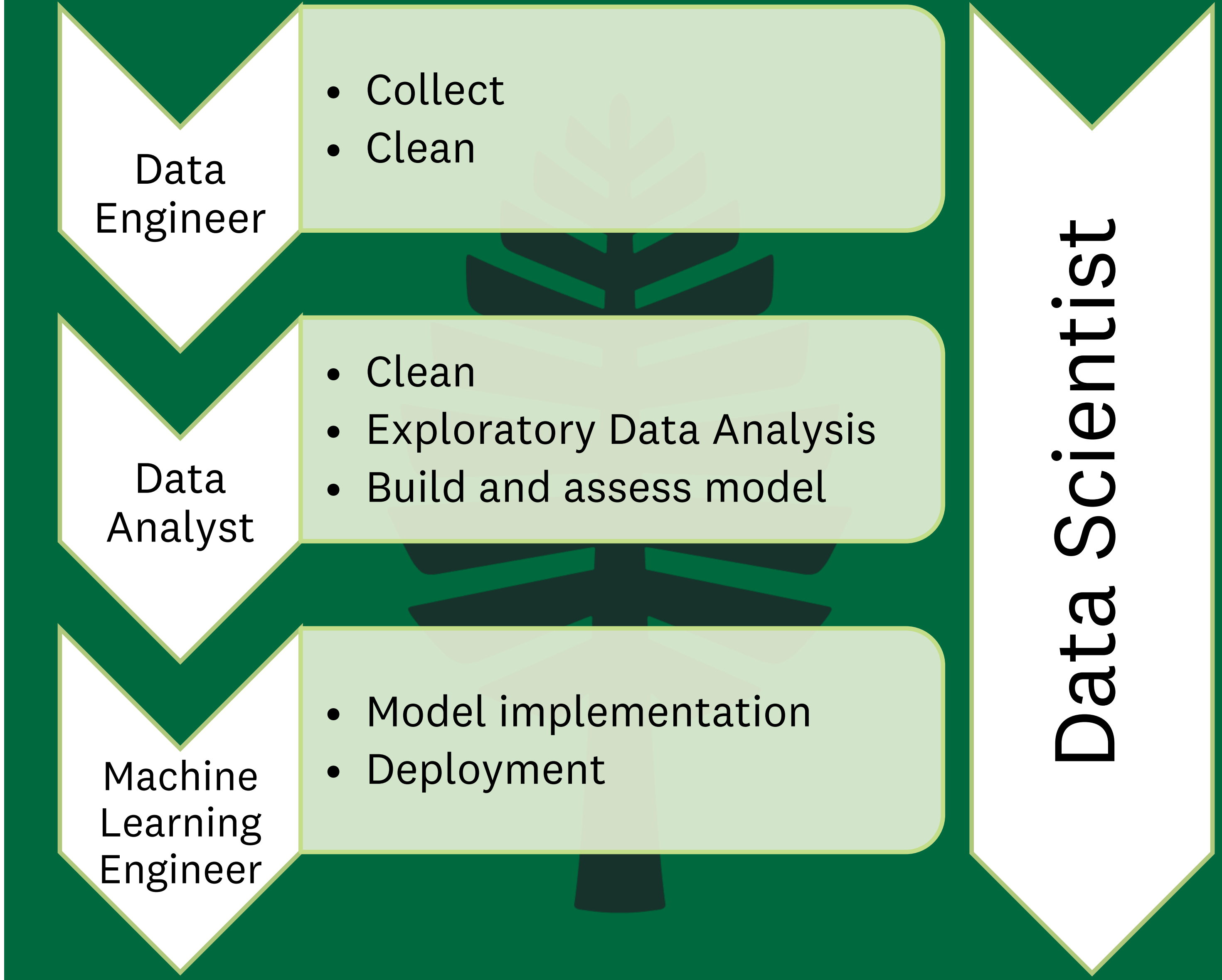
Research Data Services

Dartmouth College Library

Introduction
Data Science

Data Science is OSEMN!*

- 📥 Obtain
- 🧼 Scrub
- 🔍 Explore
- 🤖 Model
- ⁉️ iNterpret



Data Engineer
- Collect
- Clean

Data Analyst
- Clean
- Exploratory Data Analysis
- Build and assess model

Machine Learning Engineer
- Model implementation
- Deployment

Data Scientist

*pronounced “awesome” - /ˈɔ.səm/
https://www.datascience-pm.com/osemn

Collaboration on GitHub

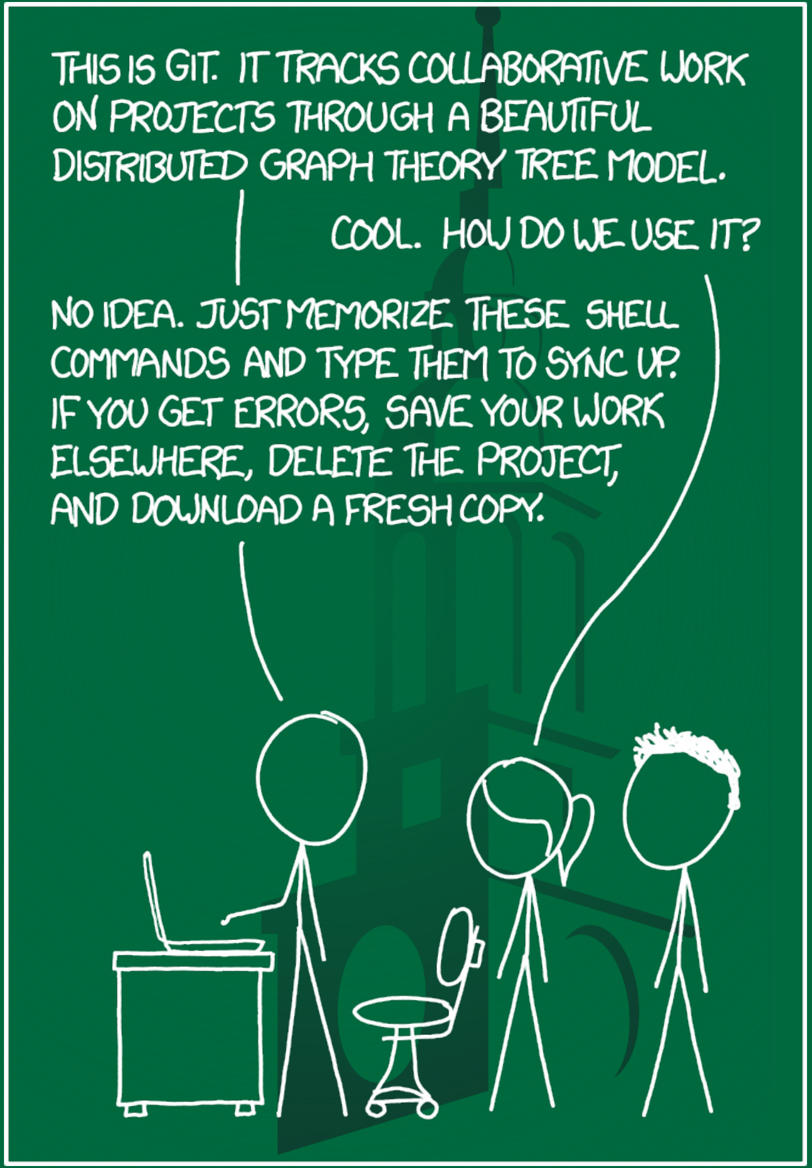
Outline

1. Introduction to GitHub
2. Setting Up Collaboration
3. Collaborative Workflows
4. Managing Collaborative
5. Further resources

Caveat Emptor

Most of today's content is from a general software developer's perspective.

Not everything may fit a Data Scientist's needs or use cases all the time.


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Munroe, R. Git. *XKCD.*
Retrieved August 3, 2023, from https://xkcd.com/1597.

Collaboration on GitHub

1. Introduction to GitHub

Git:

- 🎚️ Distributed version control system
- 💻 Lives and works on your local machine
- ☝️ Responsible for tracking changes and managing branches

GitHub:

- ☁️ Cloud-based hosting platform for Git repositories
- 🌎 Allows remote collaboration
- 📋 Offers tools for project management, collaboration, code sharing, and more

Collaboration on GitHub

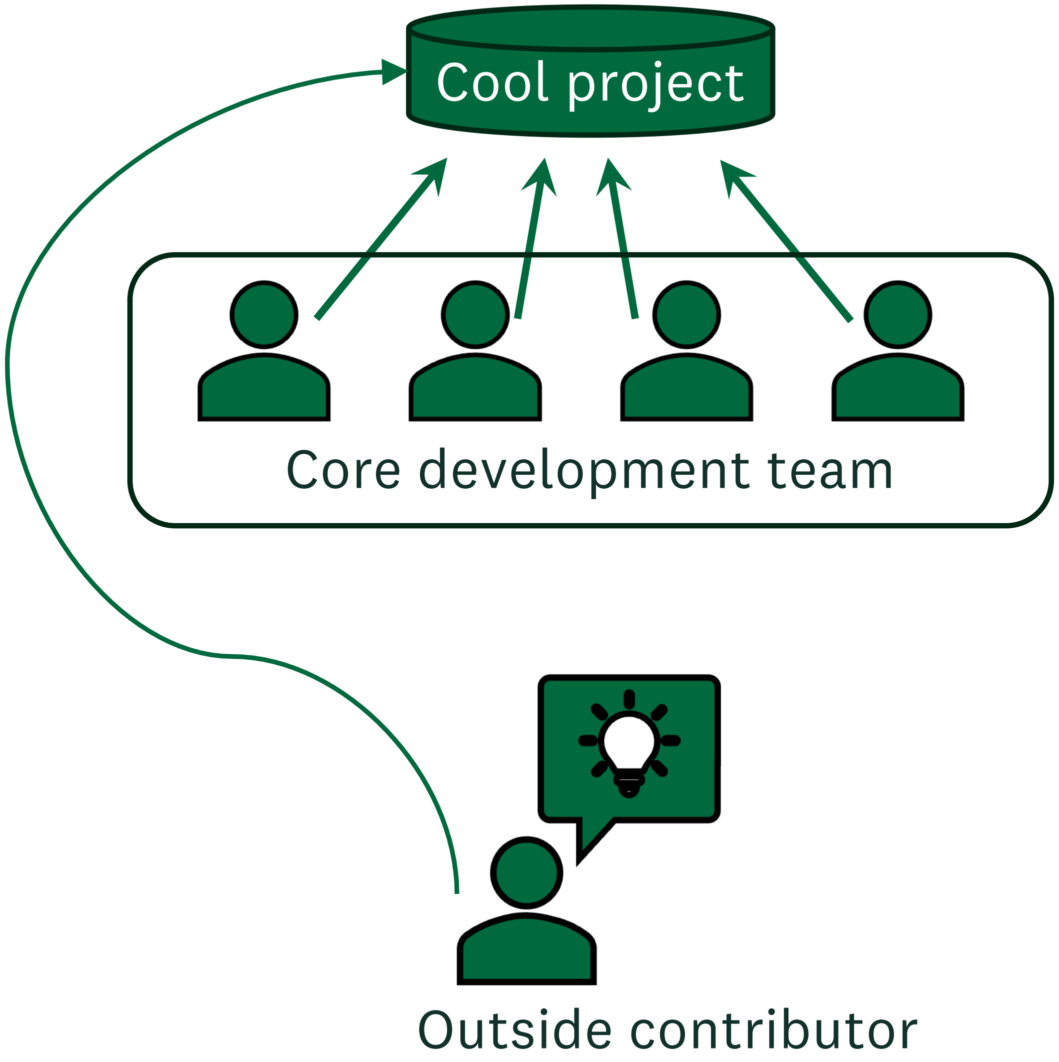
2. Setting up collaboration

Prerequisites:

- Every collaborator has a GitHub account
- Every collaborator has Git installed on their local machine

Two paradigms of collaboration:

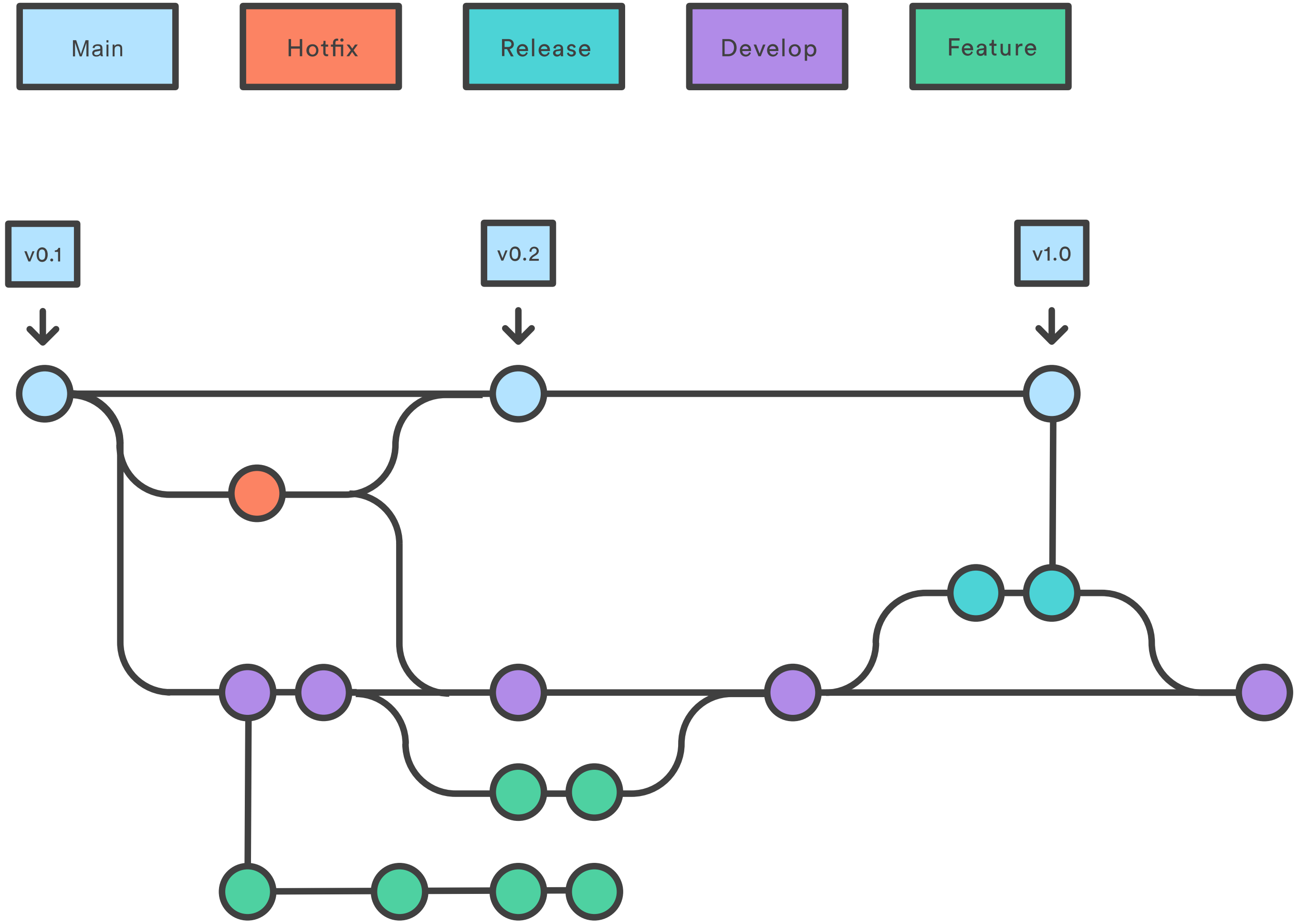
1. Being part of a team working on a shared code base
2. Contributing to another team's code base



Collaboration on GitHub
3. Collaborative Workflows

The Git Flow:

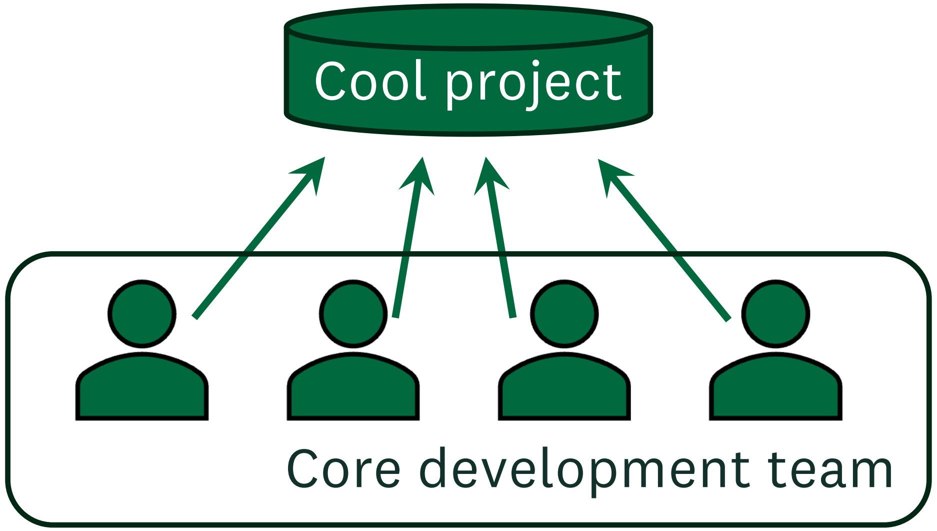
- A branch-based model
- Meaningful branching makes for a clean and informative history at different levels



3. Collaborative Workflows

Working as a team on a shared code base

- The project lives in a single repository hosted on GitHub
- All team members are added as collaborators
- Each team member works on their own branch
- When work is finished, a *pull request* is opened to request inclusion of the changes into the main branch



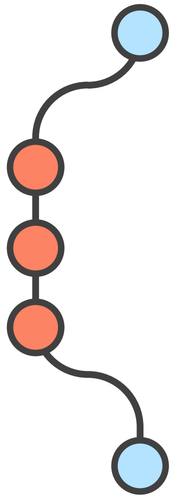
3. Collaborative Workflows

Pull requests

- A pull request is the request to merge a source branch to a target branch

Example:

> You have been working on the branch "`cool_new_feature`". After a couple of commits to this branch, the feature is now ready to be published. You initiate a pull request from "`cool_new_feature`" to "`main`"

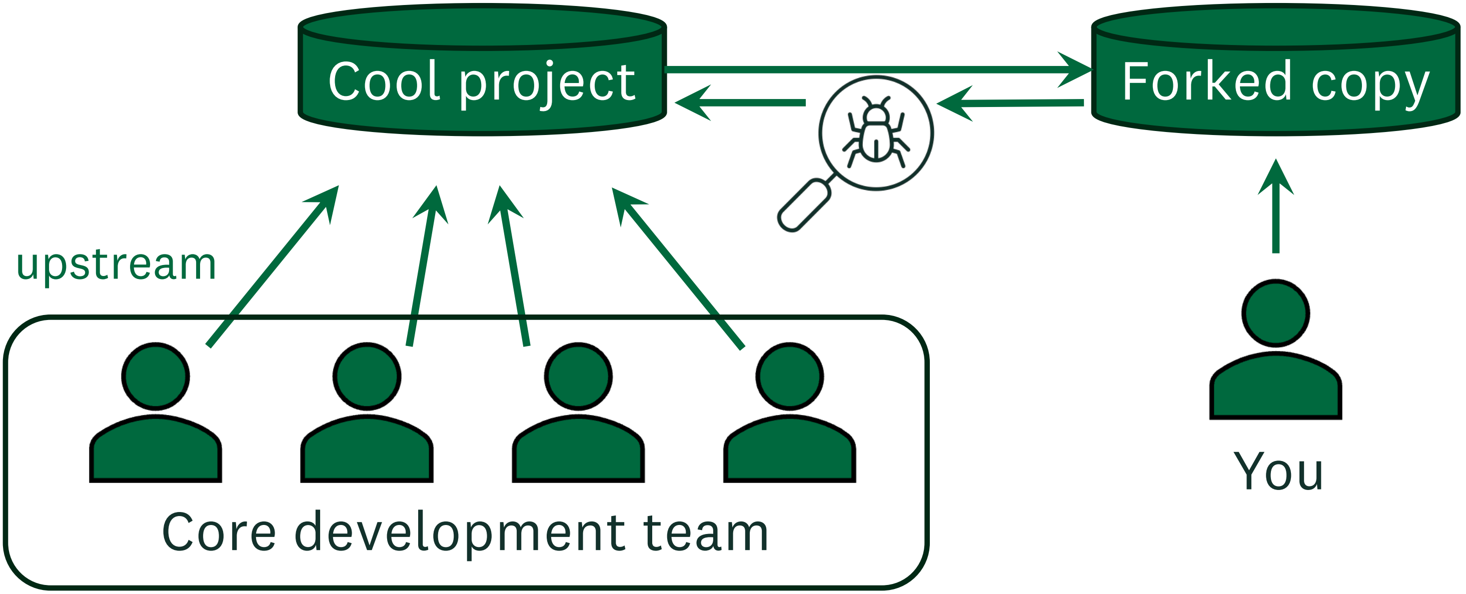


- A pull request helps to adhere to the four-eyes principle
- Pull requests are usually reviewed and often changes are requested before they are accepted (see later in this session)

3. Collaborative Workflows

Contributing to another team's code base

- Create a copy of the original project in your GitHub account (a ”fork”)
- You can always pull changes from the upstream repo
- The fork is your repo and you have full control over it
- You can ask for changes in your fork to be pulled into the original project (“pull request”)



4. Collaborative Workflows

Managing Collaborative Projects

🚩 Issues

- Flag issues
- Plan and track a chunk of work

📝 Pull request reviews

- Comment or approve proposed changes in a pull request
- Request further changes before merging

📋 Project boards

- Plan and track work
- Assign tasks
- Tasks turn into issues

💬 Discussions

- A messageboard attached to your project
- Possible references to issues, pull requests, files, …

5. Further resources

- 📚 GitHub Docs: https://docs.github.com/
- 👌 GitHub integration in VS Code: https://code.visualstudio.com/docs/sourcecontrol/github
- 🤖 Automation (CI/CD): https://github.com/features/actions

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Munroe, R. Git Commit. *XKCD*.
Retrieved August 8, 2023, from https://xkcd.com/1296.

Assignment: Peer-review

Review phase:

1. Share a link to your GitHub repo with me by EOD (simon.stone@dartmouth.edu)
2. I will share the list of links with the class via email tomorrow (August 10)
3. Each of you should go over the other repositories and reproduce the respective analyses
4. Report any bugs or suggested changes as issues in the repo's page on GitHub by **August 16**
5. Your grade will be partially based on participation in this phase

Revision phase:

1. Engage with the issues flagged in your respective project repository
2. Push changes addressing all fixable issues
3. Address all issues you do not fix by commenting on them, explaining why you do not fix them, and close them
4. Revision phase ends **August 23** (same day as presentations)
5. Your grade will be partially based on your effort to address the flagged issues in these phase

Thank you.