

More git and Github

1. Accept the organization invitation for STAT360 in your email

what you
need to write



what people
like to read

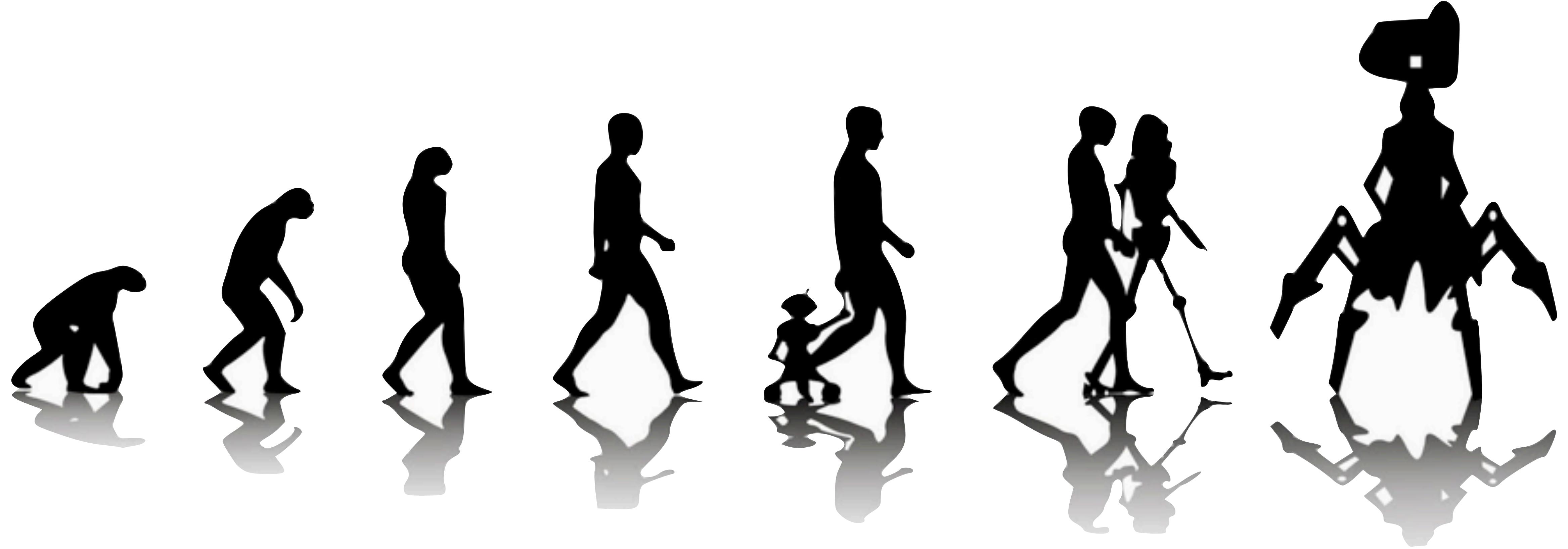
foo.R
foo.Rmd

foo.md
foo.html

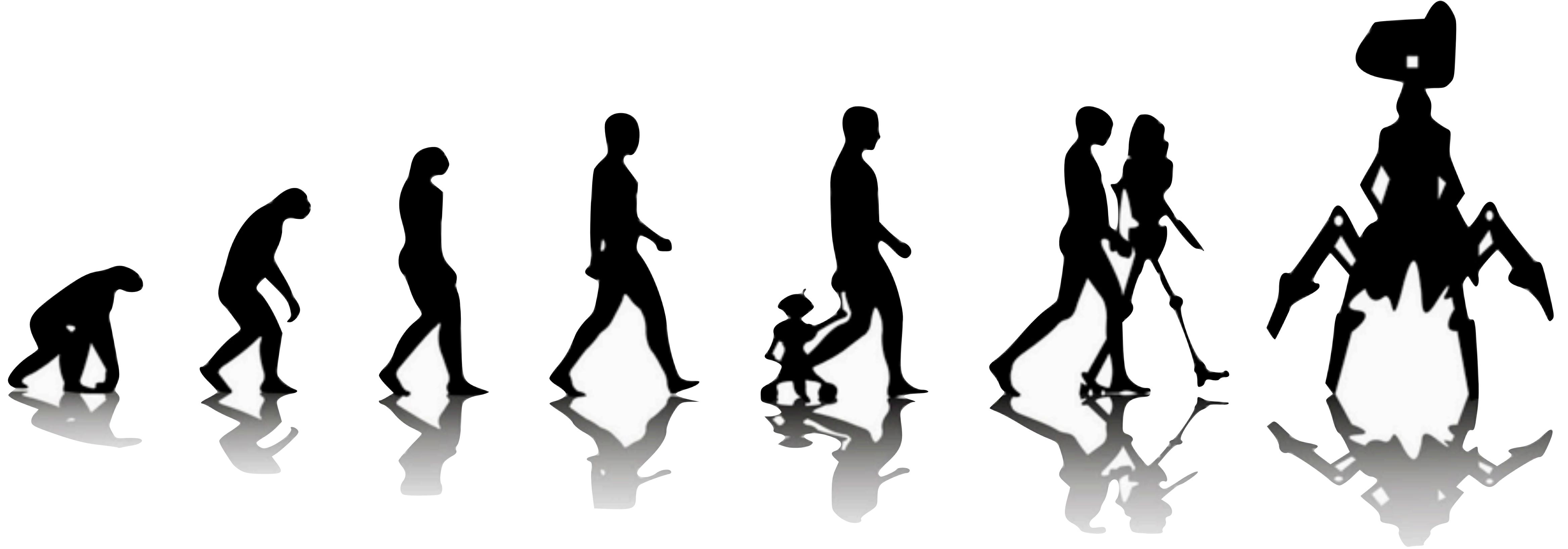


Deep Thoughts

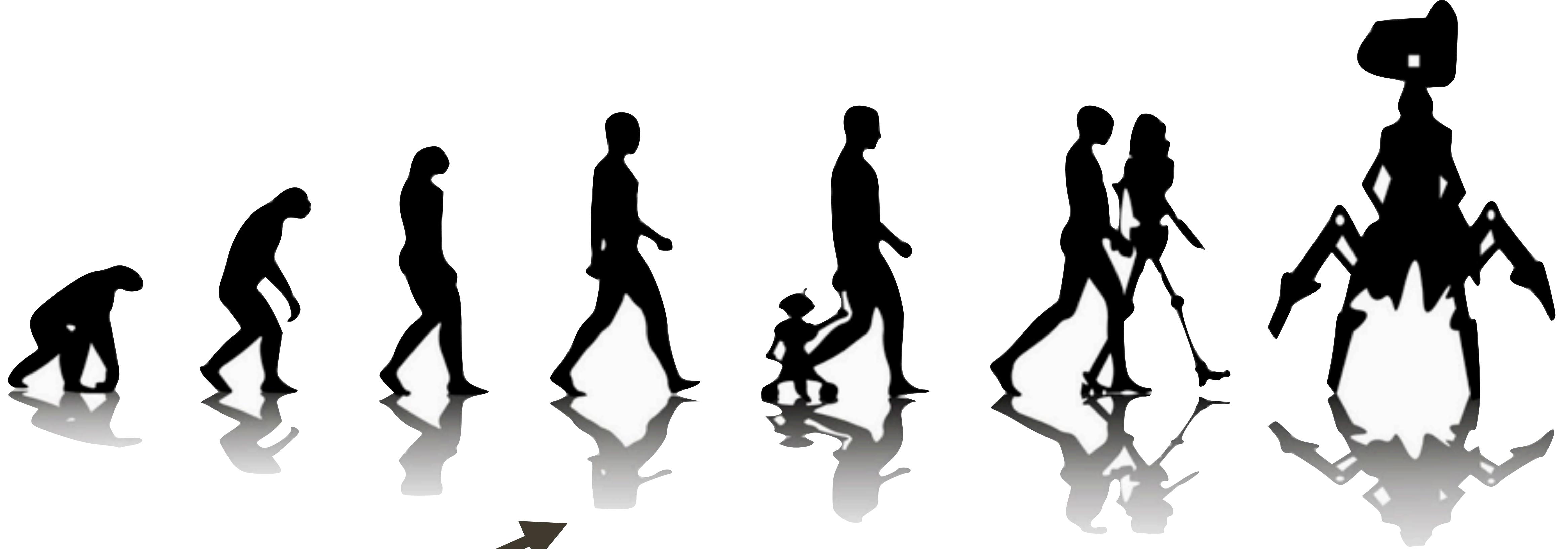




use version control

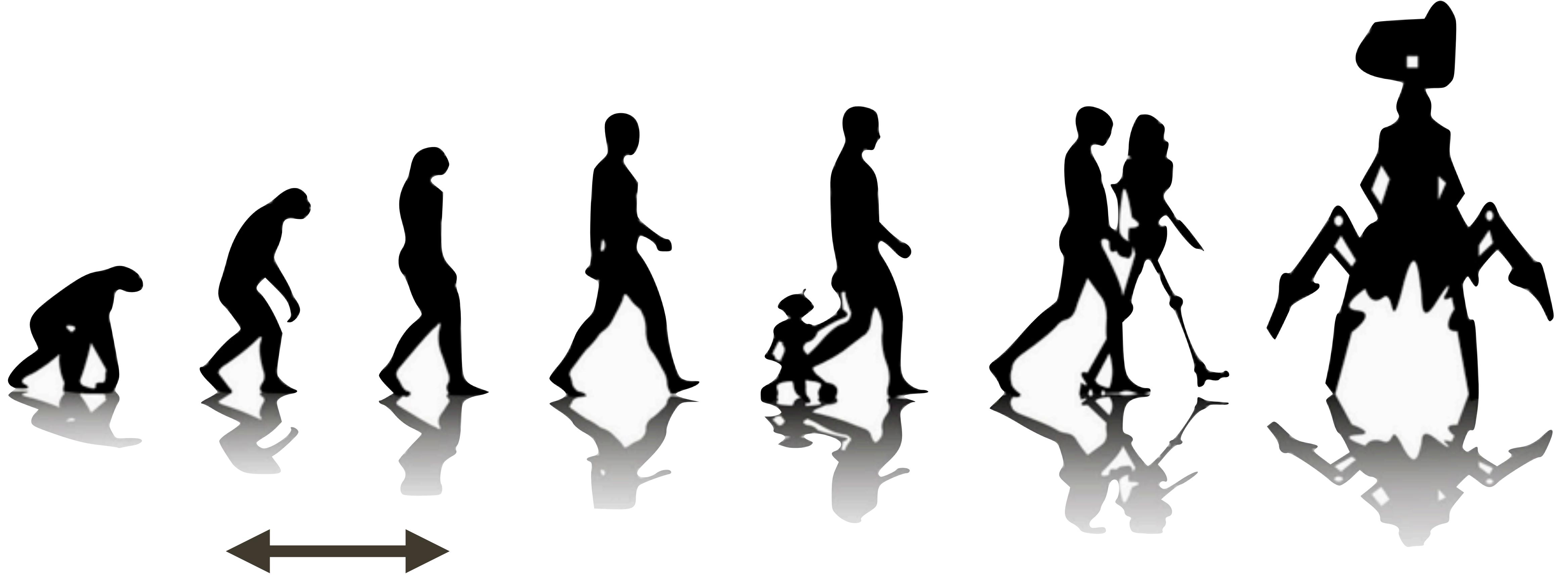


we teach Git + GitHub



"commit"

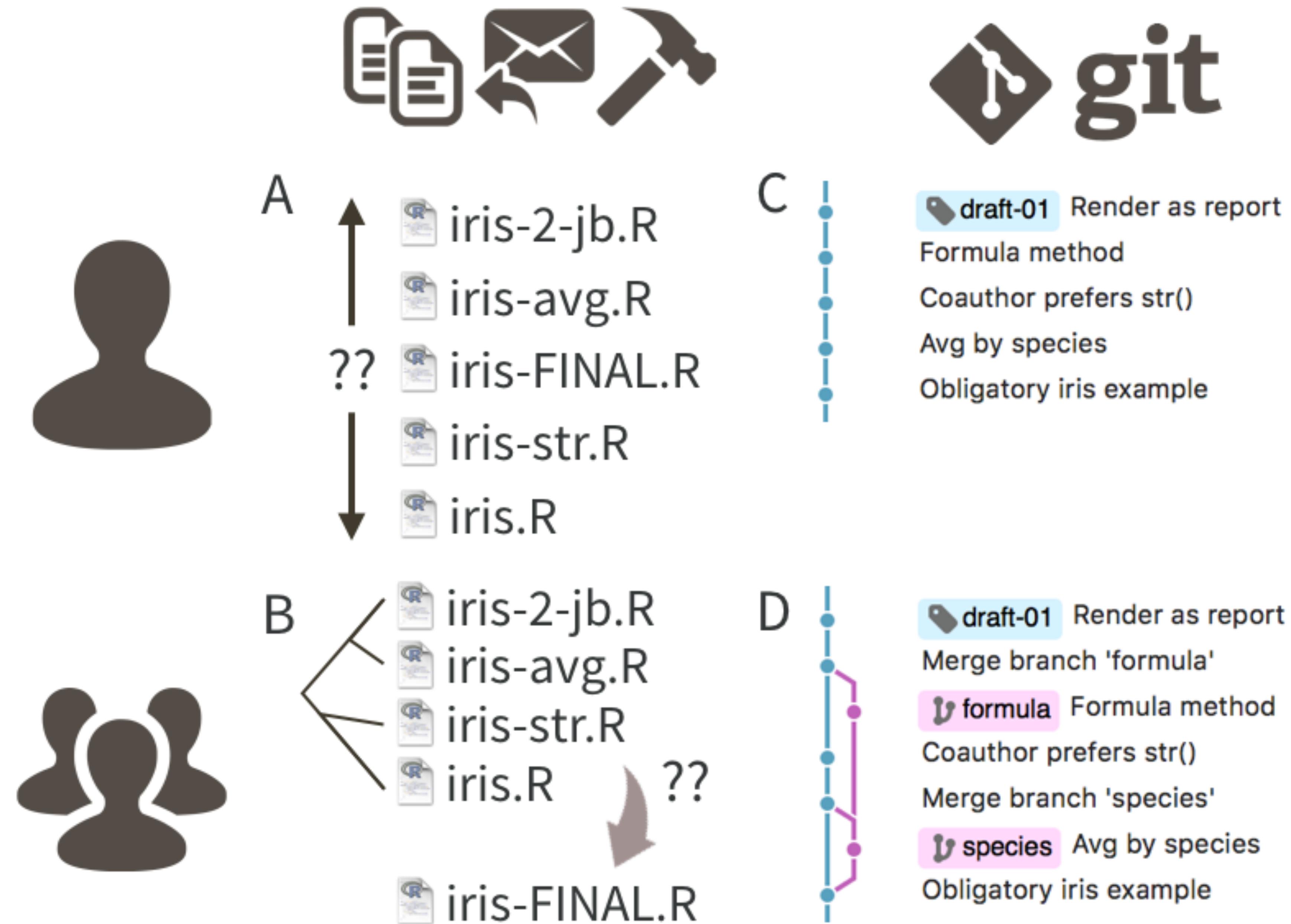
a file or project state that is **meaningful to you**
for inspection, comparison, restoration

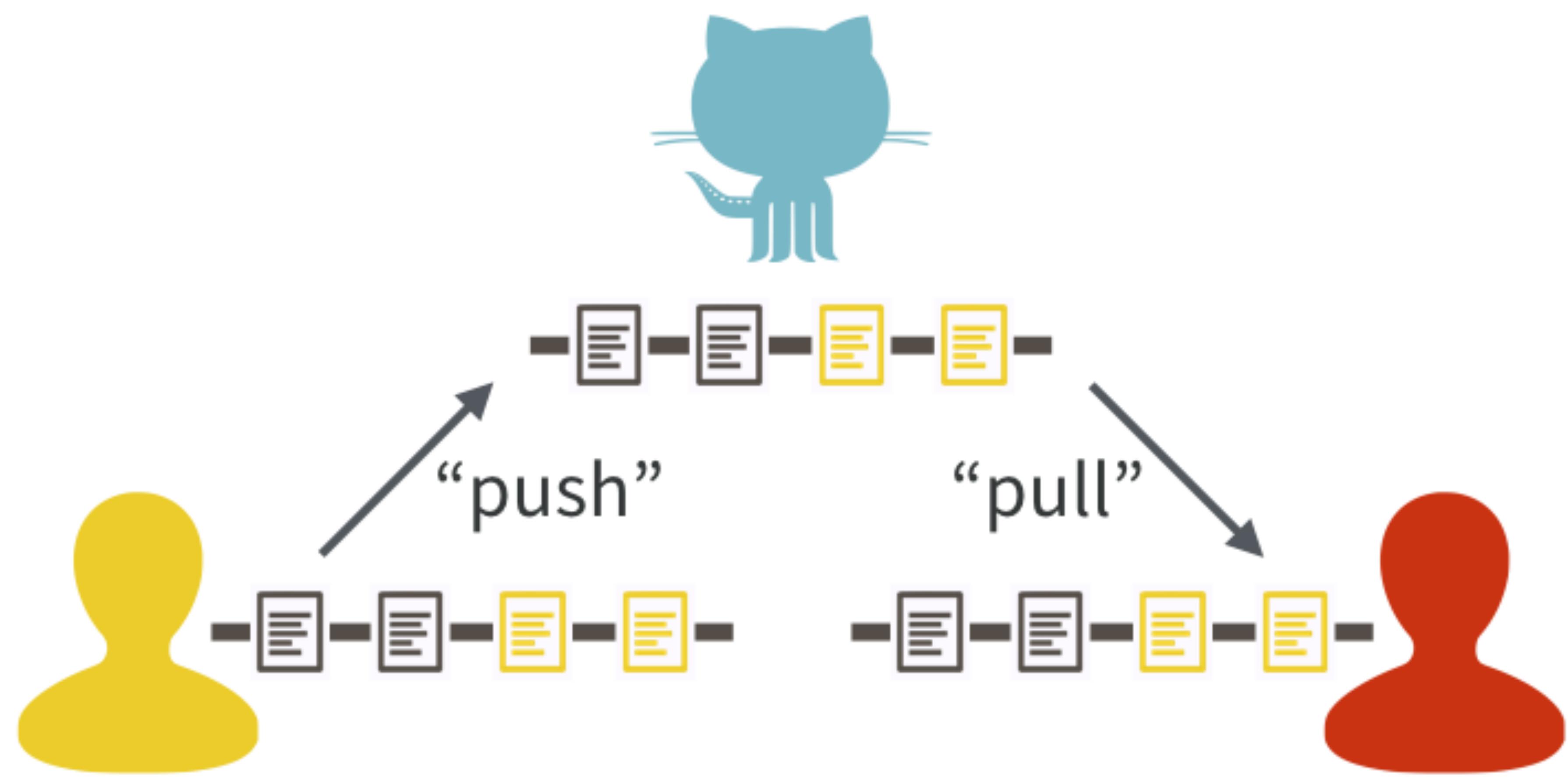


Δ

"diff"

What changed here?
Why?





collaboration

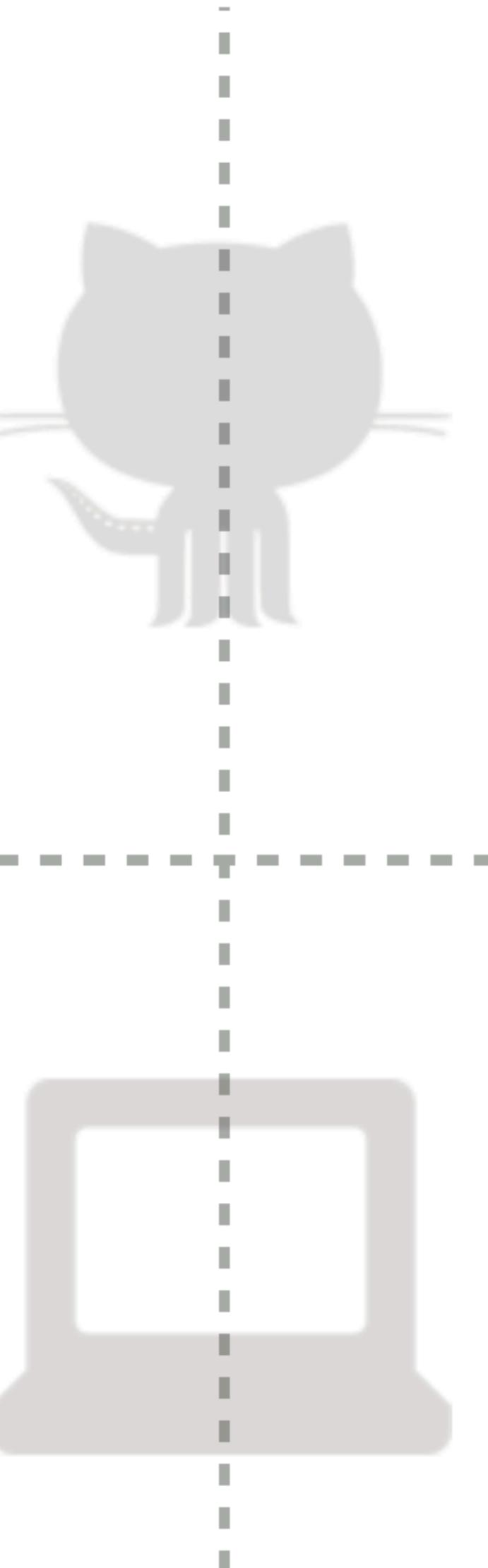
Them



You



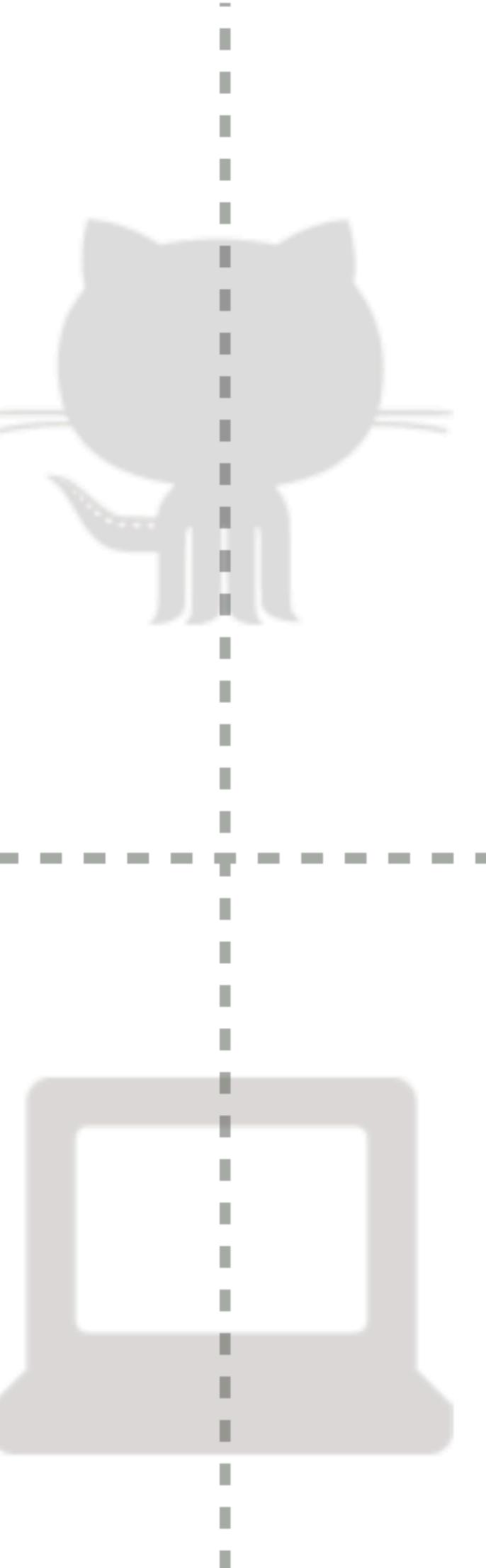
not your
problem



Them

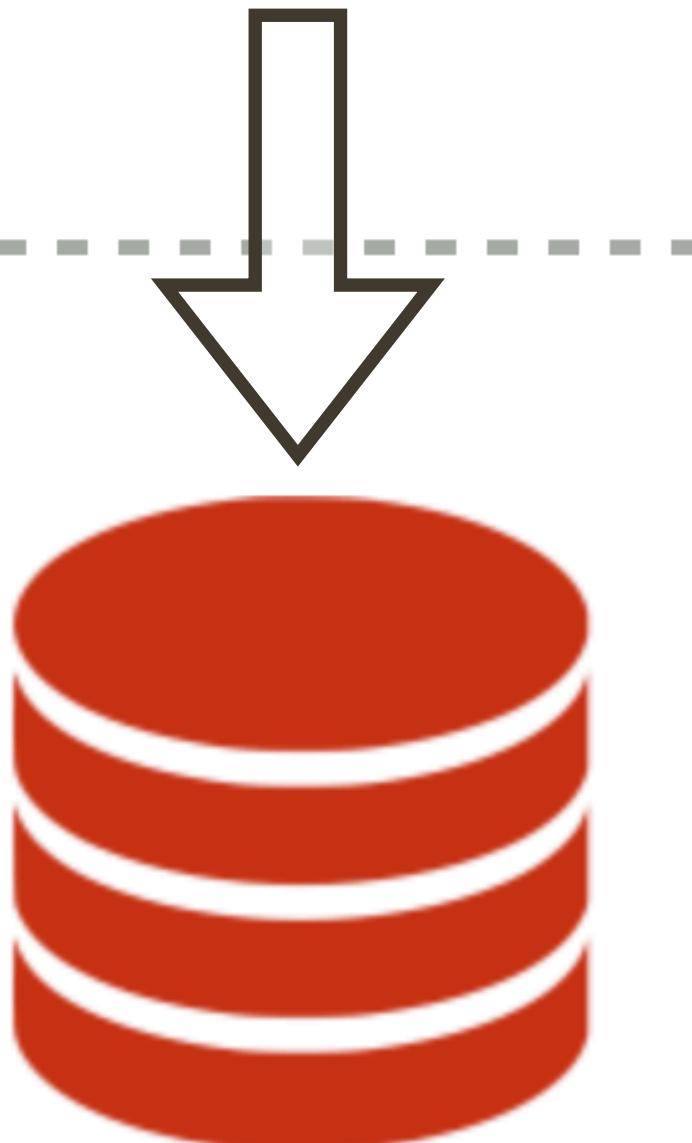


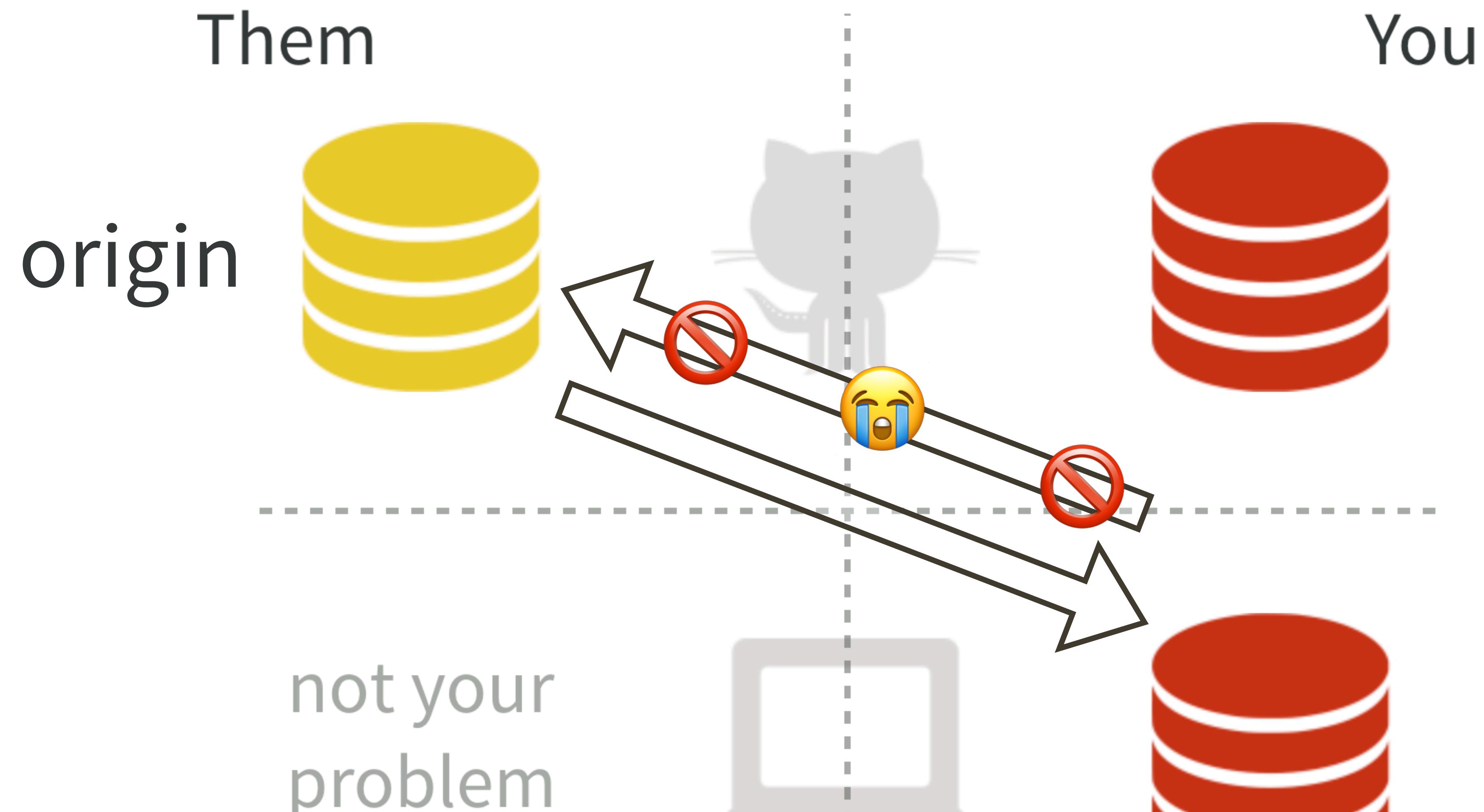
You



not your
problem

"clone"





not your
problem

"clone"

*not as useful as you might think
because you can never send a PR

Them



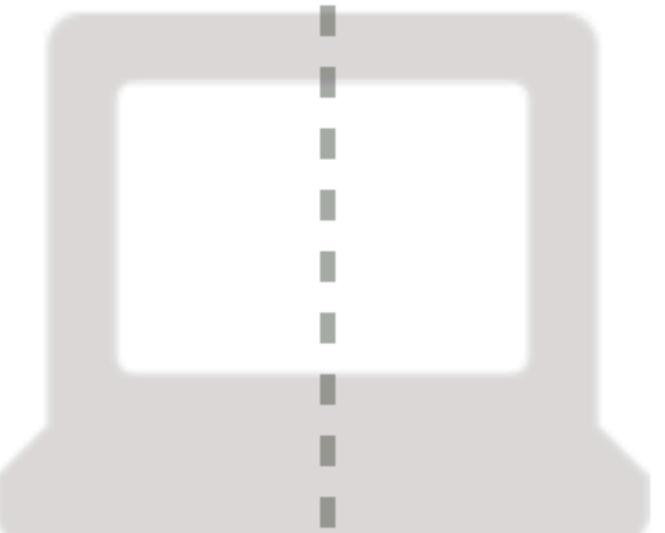
You



push ↑
↓ pull



not your
problem



daily work, your stuff

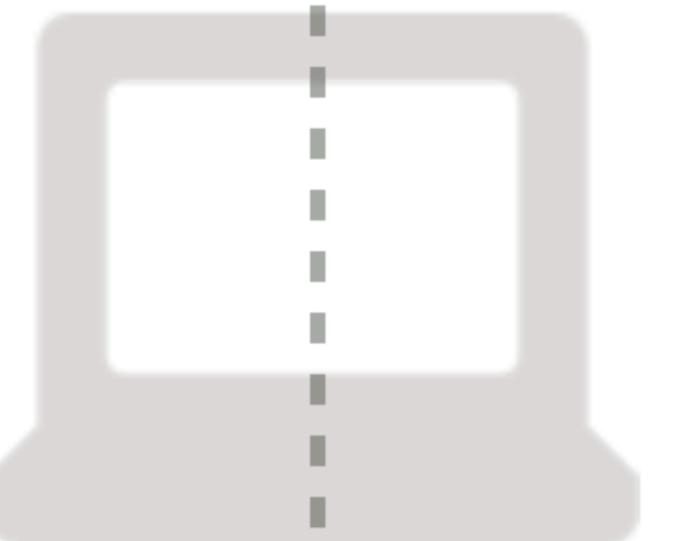
Them



You



not your
problem



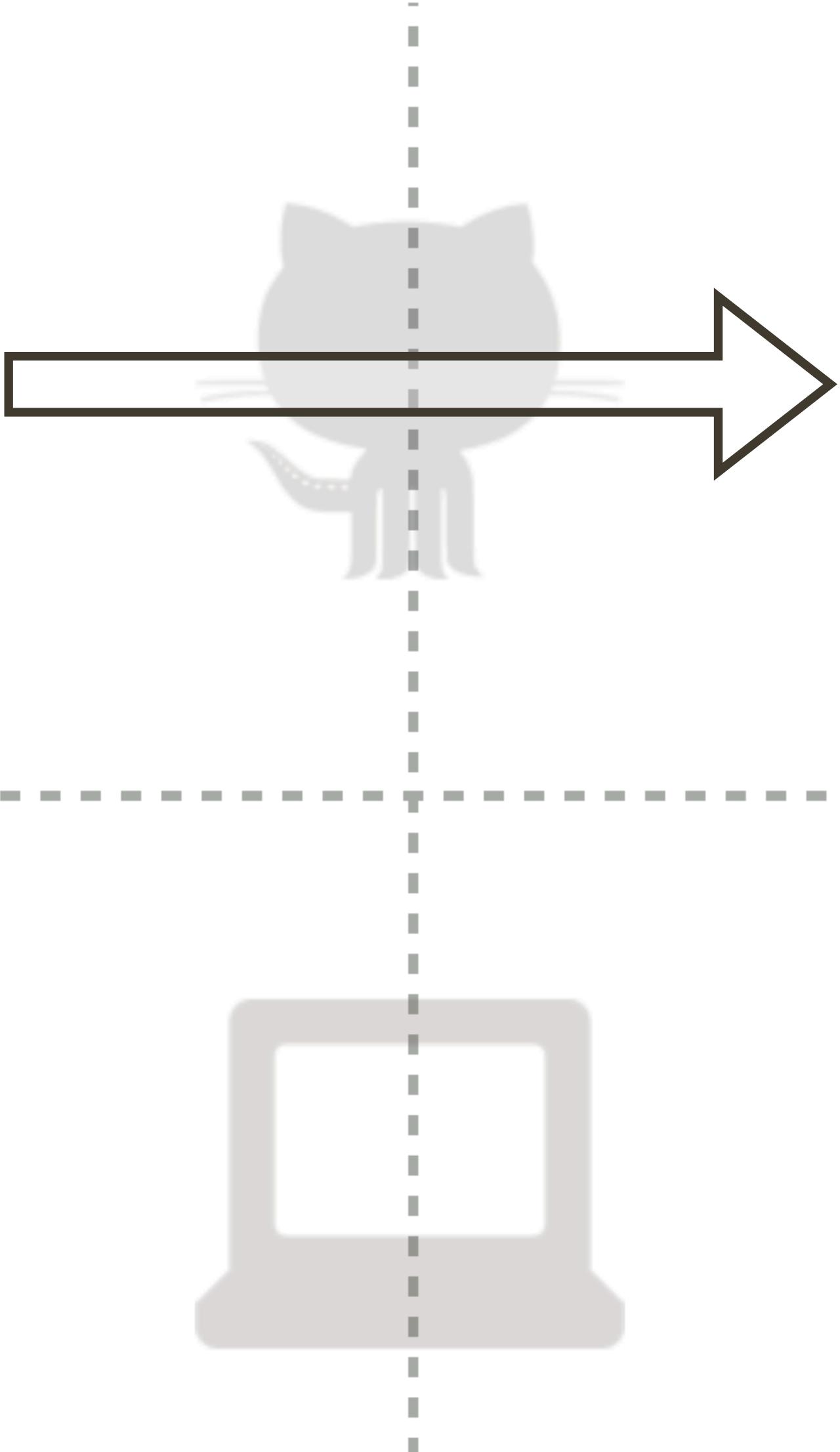
"clone"

*not as useful as you might think

Them



You



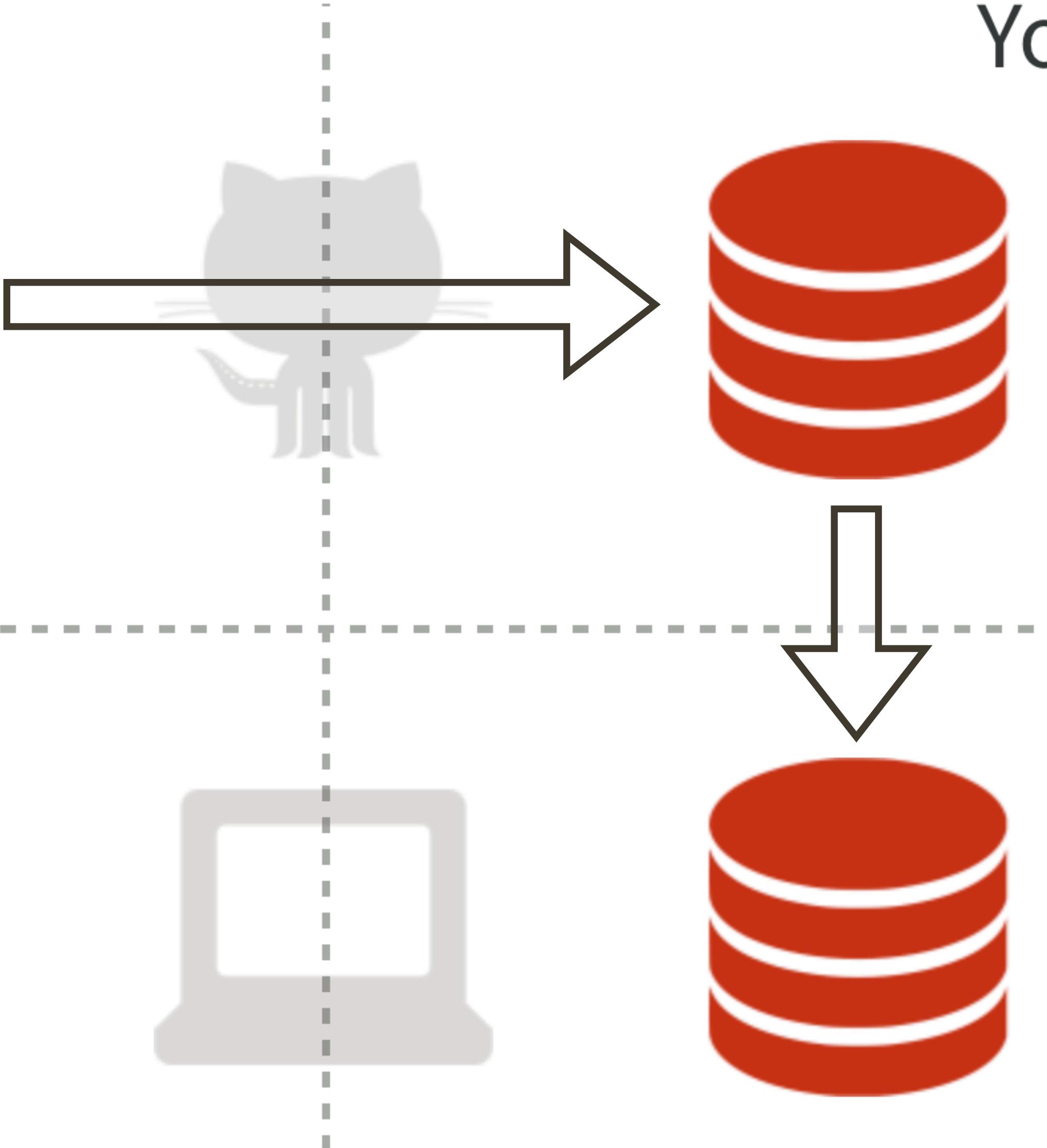
not your
problem

"fork"

Them



You



"fork and clone"

Them

????



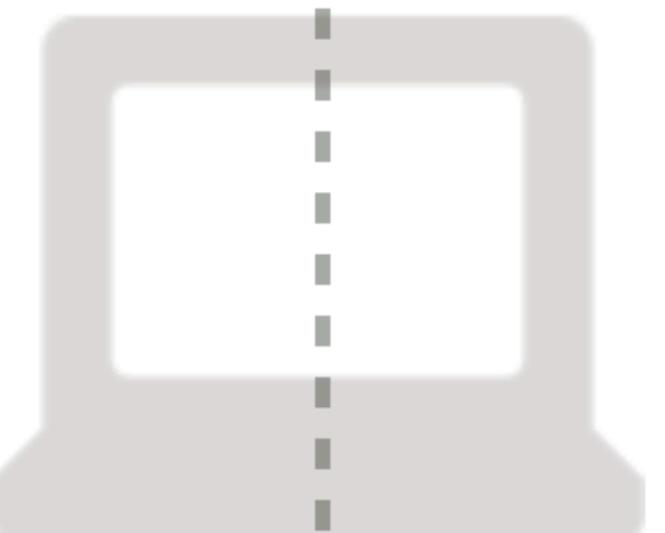
pull request

You



origin

not your
problem



push

get changes from the main repo

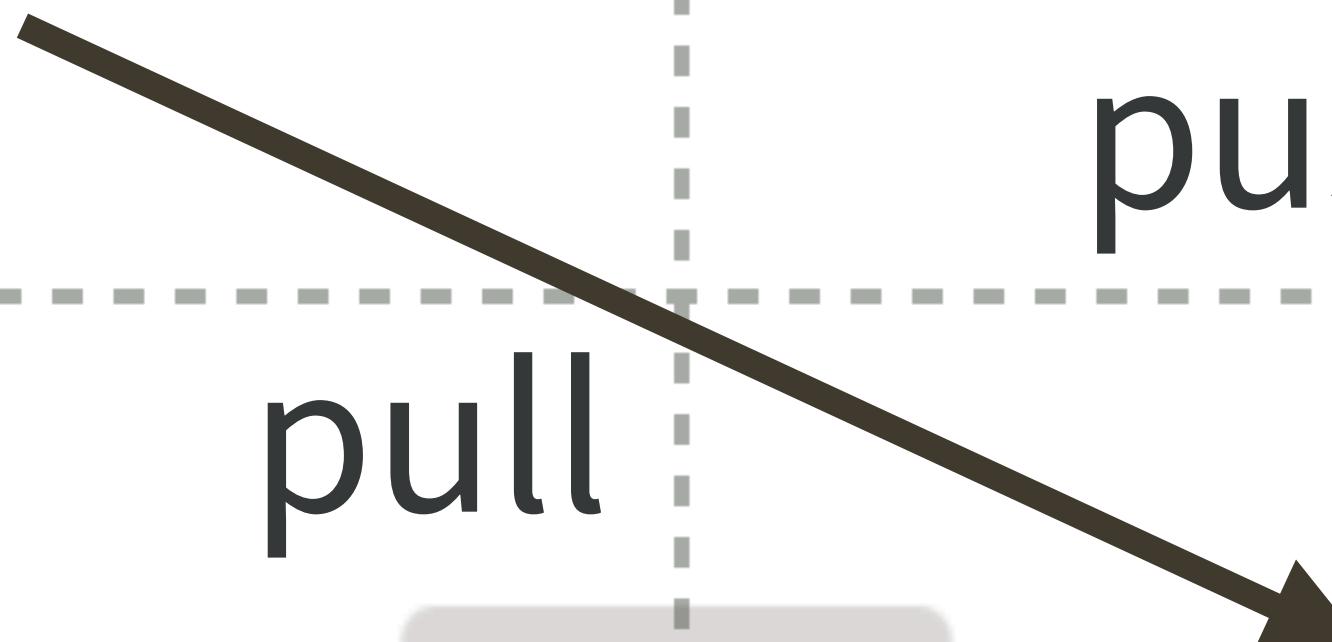
Them



You



pull request



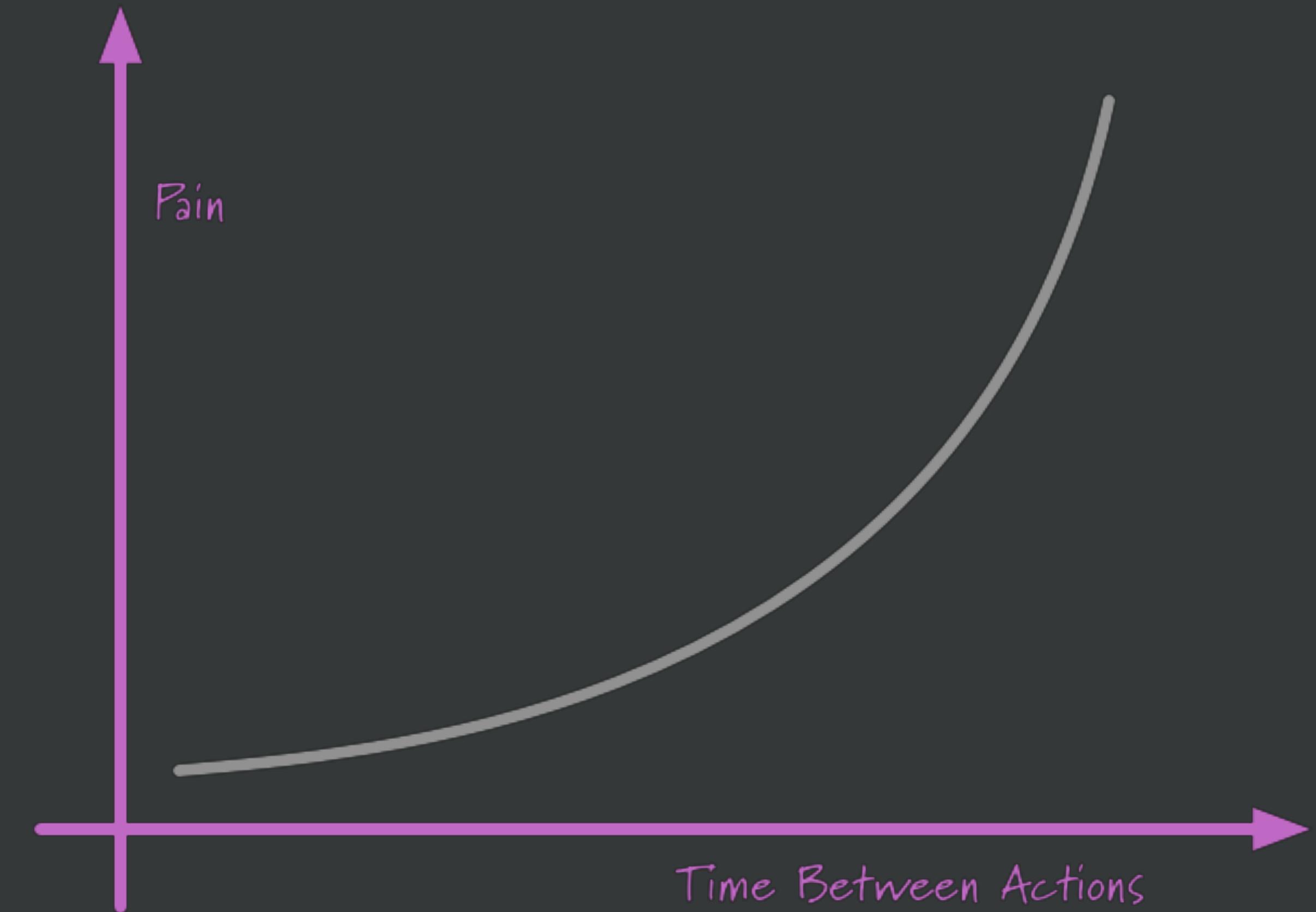
not your
problem

contribute to other people's stuff

Deep Thoughts



"If it hurts, do it more often."



<https://martinfowler.com/bliki/FrequencyReducesDifficulty.html>

"If it hurts, do it more often."

Apply this to git commit, pull, merge, push.
(and restarting R, re-running your scripts)

Why?

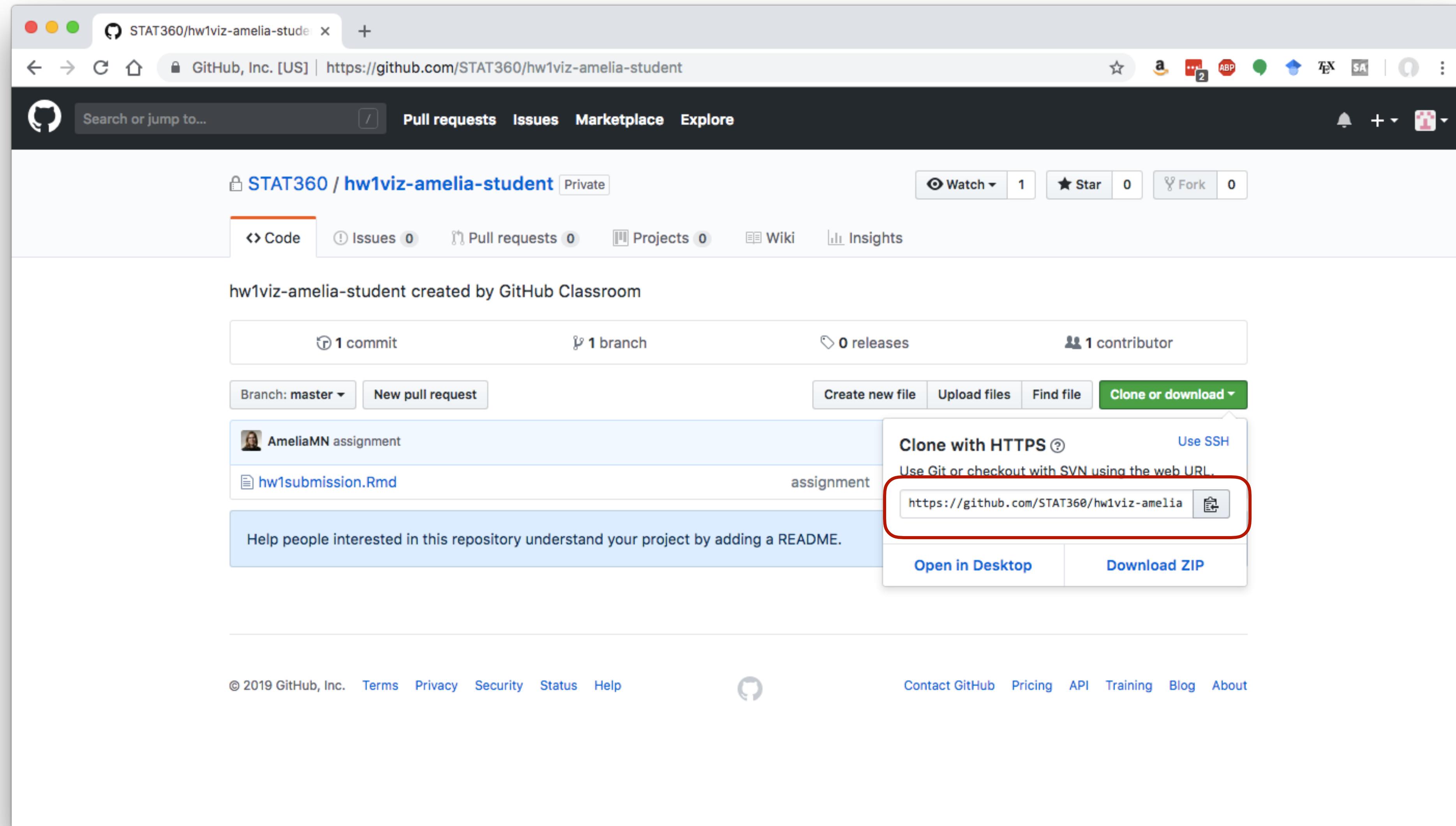
Take your pain in smaller pieces.

Tight feedback loop can reduce absolute pain.

Practice changes what you find painful.

1. Accept the organization invitation for STAT360 in your email
2. Go to github.com/STAT360 and create a team (??)
3. One (??) person from your group goes to <https://classroom.github.com/g/5uxcrcBk> to initialize the repo (fork)
4. From rstudio.cloud, create new project from version control, copy the git url like last time

Copy this URL



3. Make a new RStudio Cloud Project "from Git Repo"

The screenshot shows the RStudio Cloud interface for a space named "STAT 360". The "Projects" tab is selected. On the left, there are two existing projects: "Hw1_visualization" (created by Amelia Student on Feb 10, 2019) and "Untitled Project" (created by Amelia Student on Feb 12, 2019). A modal window titled "New Project" is open, with a red box highlighting the "New Project from Git Repo" option. The modal also includes a search bar and filter options for "List Projects" (All, Shared with everyone, Yours) and "Sort Projects" (By name, By date created).

RStudio Cloud

https://rstudio.cloud/spaces/9069/projects

STAT 360 Projects Members Info

All Projects

Hw1_visualization

Amelia Student

Created Feb 10, 2019 4:08 PM

Derived from: Hw1_visualization by Amelia McNamara

CONTINUE Hw1_visualization

AM Amelia McNamara

Created Feb 5, 2019 11:43 AM View 1 derived project ...

Untitled Project

AS Amelia Student

Created Feb 12, 2019 12:41 PM

New Project

+ New Project

New Project from Git Repo

Options

Search Projects

List Projects

All

Shared with everyone

Yours

Sort Projects

By name

By date created

RStudio Cloud

https://rstudio.cloud/spaces/9069/projects

STAT 360 Projects Members Info

All Projects

Hw1_visualization
Amelia Student
Created Feb 10, 2019 4:08 PM
Derived from: Hw1_visualization
CONTINUE Hw1_visualization
Amelia McNamara
Created Feb 5, 2019 11:43 AM View 1 derived project ...

New Project from Git Repo

URL of your Git repository

https://github.com/STAT360/hw1viz-amelia-student.git

Packages

Add packages from the base project

OK

Options

Search Projects

List Projects

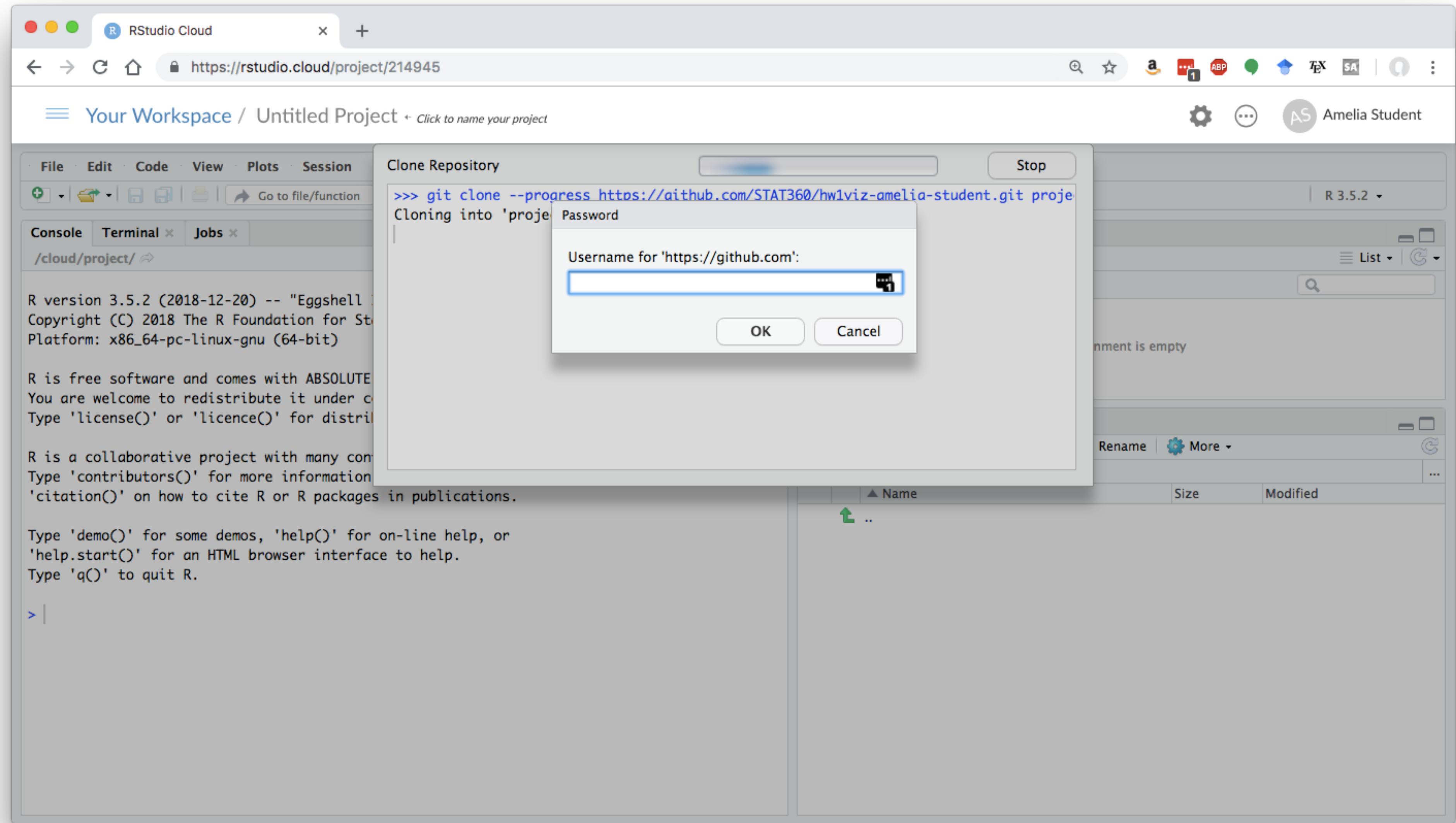
All Shared with everyone Yours

Sort Projects

By name By date created

The screenshot shows the RStudio Cloud interface for a STAT 360 workspace. On the left, there's a list of projects: 'Hw1_visualization' by 'Amelia Student' (created Feb 10, 2019) and 'Hw1_visualization' by 'Amelia McNamara' (created Feb 5, 2019). A modal window titled 'New Project from Git Repo' is open in the center. It contains a text input field with the URL 'https://github.com/STAT360/hw1viz-amelia-student.git'. Below it is a 'Packages' section with a checked checkbox for 'Add packages from the base project'. At the bottom right of the modal is a blue 'OK' button. To the right of the modal, there are several sidebar options: 'Options' (with a search bar), 'List Projects' (radio buttons for 'All', 'Shared with everyone', and 'Yours'), and 'Sort Projects' (radio buttons for 'By name' and 'By date created'). The top navigation bar includes links for 'Projects', 'Members', and 'Info', along with user profile icons for 'Amelia Student' and 'Amelia McNamara'.

Probably— enter Github credentials



Introduce yourself to git

```
git config --global user.name 'Jane Doe'  
git config --global user.email 'jane@example.com'  
git config --global credential.helper 'cache --timeout 3600'  
git config --global --list
```

1. Accept the organization invitation for STAT360 in your email
2. Go to github.com/STAT360 and create a team (??)
3. One (??) person from your group goes to <https://classroom.github.com/g/5uxcrcBk> to initialize the repo (fork)
4. From rstudio.cloud, create new project from version control, (copy the git url like last time)
5. (??) Make a new branch
6. Have one person work from their machine and make some changes, as the rest of you watch and contribute (maybe, on the screen)
7. One person commits + pushes to Github
8. Everyone else, pull from Github

Recovering from Git(Hub) failure

Scenario: You try to push and cannot

What's the problem?

There are changes on GitHub that you don't have.

Pull. If the gods smile upon you, merge works. Now push.

Let's create this situation.

Make sure local Git pane is clear.

Make sure local and remote are synced (push, pull).

Edit & commit to file A locally.

Edit & commit to file B remotely.

Try to push. You will fail.



```
jenny@2015-mbp bunny-scarf $ git push  
To github.com:jennybc/bunny-scarf.git  
! [rejected]      master -> master (fetch first)  
error: failed to push some refs to 'git@github.com:jennybc/bunny-scarf.git'  
hint: Updates were rejected because the remote contains work that you do  
hint: not have locally. This is usually caused by another repository pushing  
hint: to the same ref. You may want to first integrate the remote changes  
hint: (e.g., 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

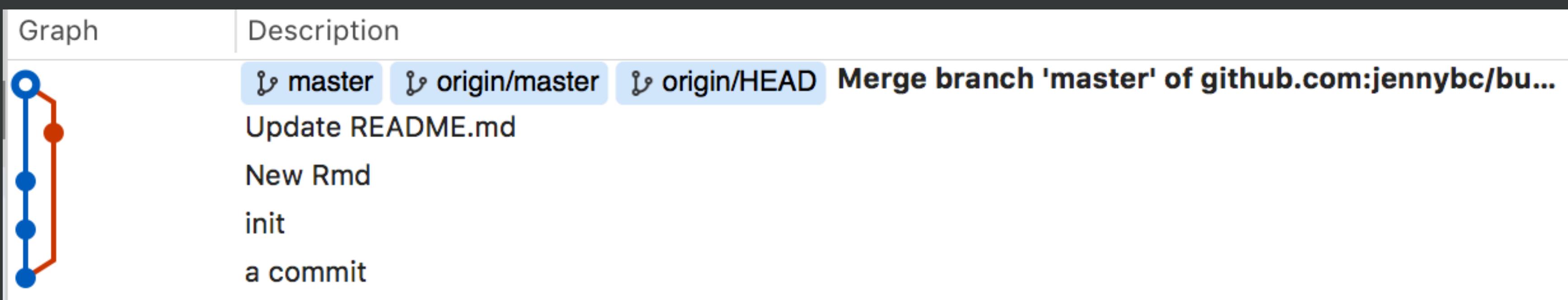
Remedy? Do what it says!

pull, then push ... **pull, then push** ... pull, then push

Look at your Git history.

You will see a merge commit, where the local and remote changes were reconciled.

This is best case scenario and is likely with good Git habits (lots of small frequent commits and merges, no binary files in repo).



Recovering from Git(Hub) failure

Scenario: You pull and get a merge conflict.

What's the problem?

GitHub can't figure out how to reconcile diffs.

Resolve the conflicts.

Or abort ... and come back later.

Let's create this situation.

Make sure local Git pane is clear.

Make sure local and remote are synced (push, pull).

Edit & commit to file A locally.

Make conflicting edit & commit to file A remotely.

Try to push. You will fail. Try to pull. You will fail. All is fail.

From `github.com:jennybc/bunny-scarf`

`958548f..3357952 master -> origin/master`

Auto-merging `README.md`

`CONFLICT (content): Merge conflict in README.md`

Automatic merge failed; fix conflicts and then commit the result.

`<<<<< HEAD`

`Wingardium Leviosaaaaaaaa`

`=====`

`Wing-GAR-dium Levi-0-sa`

`>>>>> 33579525d88af071268b0a0c64c54f357712589a`

From `github.com:jennybc/bunny-scarf`

`958548f..3357952 master -> origin/master`

Auto-merging `README.md`

`CONFLICT (content): Merge conflict in README.md`

Automatic merge failed; fix conflicts and then commit the result.

`<<<<< HEAD`

`Wingardium Leviosaaaaaaaa`

`=====`

`Wing-GAR-dium Levi-0-sa`

`>>>>> 33579525d88af071268b0a0c64c54f357712589a`